

**PROSOFT'S**



**Word Processing System  
for the TRS-80 AGAIN**

**written by**

**Chuck  
Tesler**

**Glenn  
Tesler**

**Los Angeles  
1984**



**This is not a scanned copy of ALLWRITE.**

The original version of this text was typed by Kelly Bates during 1990/1991, in Model 4 Allwrite format. I converted it from Model 4 to MSDOS using David P. Millers MS UTILITIES. The text was then formatted using Word 97 and saved in Adobe PDF format.

I tried to keep the “look-and-feel” of the original manual and at the same time shorten it from its 338 pages to something more manageable. The Index has not been included, but I did include an updated Table of Contents.

Pete Bumgardner  
TRS-80 AGAIN

This book explains how to use the ALLWRITE! Word Processing System on the TRS-80 micro-computer. It applies to ALLWRITE! Release 1.0 and above. Additional Supplement sheets may be issued from time to time.

Date: April, 1984

Printing: 5 4 3 2 1

Copyright © 1984, The Tesler Software Corporation (TTSC)

Published by: PROSOFT, a division of TTSC  
Box 560  
North Hollywood, CA 91603

All rights reserved. No part of this book may be reproduced by any means without the express written permission of the publisher. Examples are for personal use only. Every reasonable effort has been made to ensure accuracy throughout this book, but the authors and publisher assume no responsibility for any errors or omissions. No liability is assumed for damages resulting from the use of information contained herein.

“PROSOFT” and “NEWSCRIPT” are Registered Trademarks of The Tesler Software Corporation.

“ALLWRITE!” is a Trademark of The Tesler Software Corporation.

“TRS-80” and “TRSDOS” are Registered Trademarks of Tandy Corporation.

“DOSPLUS” is a Registered Trademark of Micro-Systems Software, Inc.

“Electric Webster” is a Trademark of Cornucopia Software, Inc.

Some of our disks contain “TRSDOS 6.x”, under license from Tandy Corporation, and copyrighted 1982/83/84 by Logical Systems, Inc., which has licensed it to Tandy Corporation.

Some of our disks contain “DOSPLUS” or “TDOS” under license from and copyrighted 1981/82/83/84 by Micro-Systems Software, Inc.

ALLWRITE IS PROVIDED 'AS-IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A **PARTICULAR PURPOSE.** **PROSOFT**, The Tesler Software Corporation, and/or the authors shall have no responsibility or liability to the purchaser or any other person, persons, or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by use of ALLWRITE and/or its documentation. This expressly includes, but is not limited to, loss or invalidation of customer data, programs, files, or business opportunities.

Purchase of this software conveys to CUSTOMER title to the media (the diskette) but not title to the computer programs or files. CUSTOMER is granted a non-exclusive license to use this software on ONE computer.

Copies of SOFTWARE may be made only for BACKUP purposes. If you transfer any copy or modified copy of any portion of ALLWRITE to another party, your license is automatically terminated and all support, maintenance, and update privileges are lost. If ALLWRITE is to be used on more than one computer at a time in your company, a separate copy must be purchased for each machine. Please contact us for volume pricing.

### **MEDIA WARRANTY**

**PROSOFT** warrants that the **ALLWRITE** diskette is free from mechanical or recording defects, and that we will replace any such defective diskettes within 90 days of the date of purchase. No other warranties are expressed or implied as to the operation, use or suitability of ALLWRITE.

\* \* \* \* \*

Please record below the Serial Number and Date of your copy of ALLWRITE. These are on the diskette and displayed each time ALLWRITE is started.

## **ACKNOWLEDGEMENTS**

ALLWRITE was written by Chuck Tesler and Glenn Tesler. The entire text of this book was written and printed with ALLWRITE, using a TRS-80 Model 4 and a Diablo printer with a Theme 11 wheel. If your printer has the necessary mechanical capabilities and is supported by ALLWRITE, you can do anything you see in this book.

Electric Webster was used to check and correct most of the spelling throughout this book and to generate hyphenation points in selected sections.

The cover was designed by Richard Goldstein of Quest Studios.

We must confess that the woodcut illustrations were not printed by ALLWRITE; we took them from catalogs of 19th Century artwork.

We wish to express our gratitude to the people who helped us test ALL WRITE for several months prior to its release; to the people who patiently read and critiqued this book; and to several thousand NEWSCRIPT users whose suggestions over the past three years guided the development of ALLWRITE.

Special thanks are due to Ron Malo and Dino Sanchez for their comments, suggestions, and bravery in testing and reading the earliest copies of ALLWRITE... before it was all right.

# TABLE OF CONTENTS

CHAPTER 1 -- INTRODUCTION .....	01
How to use this book .....	01
Equipment requirements .....	02
Operating Systems .....	03
Software Compatibility .....	03
Customer Support .....	04
Reporting Problems .....	04
Service Charges .....	05
Updates .....	05
Terminology .....	06
Notation .....	07
CHAPTER 2 -- INSTALLATION .....	09
Contents of the ALLWRITE package .....	09
Outline of procedure .....	09
Reference (Cue) Cards .....	10
Backup copies .....	10
Format of the distribution disk .....	11
What's on the Distribution disk .....	11
Computer Dependencies .....	12
Copying the distribution disk(BACKUP) .....	13
Building a working disk .....	13
Tailoring ALLWRITE to your req'mts .....	14
Printer selection .....	15
Printer Attachment .....	15
Serial (RS-232) printers .....	15
print spoolers .....	16
Cut (single) sheets .....	16
Automatic line feeds .....	16
Multiple printers .....	17
Screen and cursor selection .....	17
Auto Save .....	17
Verifying Disk Writes .....	18
Audible click .....	18
Keyboard drivers .....	18
Operating Systems (DOS) .....	19
Finishing installation .....	19
Daisywheel printers .....	19
Automatic startup .....	20
Printer description tables .....	20
Re-running installation .....	20
One-drive systems .....	20

## CHAPTER 2 -- Continued

Data disks .....	21
Printer switch settings .....	21
Converting NEWSCRIPT files .....	21
Differences between NEWSCRIPT / ALLWRITE .....	23
Using NEWSCRIPT Conventions .....	24
Control word changes .....	25
Mailing labels .....	26
Daisywheel proportional .....	26
Electric Webster .....	26
Dotwriter compatibility .....	26

## CHAPTER 3 -- Learning to use ALLWRITE

Outline of the chapter .....	27
Starting ALLWRITE .....	29
Entering text with the editor .....	29
Typeahead .....	30
Moving the cursor with arrows .....	31
Scrolling .....	31
Overstriking text .....	31
Quick cursor movements .....	32
The Control key .....	32
Deleting text .....	32
Inserting text .....	33
Inserting a single character .....	34
Top and end of file .....	34
Command mode .....	34
STATUS of text .....	34
Screen print .....	36
WHOOPS - discarding typos .....	36
SAVE text to disk .....	36
On-line HELP .....	36
DIRECTORY of a disk .....	37
Exiting from ALLWRITE - QUIT .....	38
Editing Another File - NEW .....	38
Formatted printing .....	38
On-screen preview formatting .....	39
Control word rules .....	39
Fundamental Control Words .....	40
Fundamental Emphasis Marks .....	40
Sample letter .....	40
Printing a letter .....	42
Previewing printing .....	43
Formatting errors .....	44

## CHAPTER 4 -- THE TEXT EDITOR

Topics in this chapter .....	47
Recovering text after system failure .....	48
Quick start of the Editor .....	49

## CHAPTER 4 -- Continued

Block operations .....	49
Block copy .....	50
Block move .....	50
Block delete .....	50
Listing(printing)a block as-is .....	50
Block putfile .....	51
Getting text from disk .....	52
Copying text between files .....	53
Search and Replace .....	53
SEARCH command .....	54
SEARCH backwards .....	55
Approximate (masked) search .....	55
Search control key .....	55
Replacing text .....	56
REPLACE Control key .....	56
Search-and-Replace .....	57
Global Search-and-Replace .....	57
CHANGE command .....	57
How changes are counted .....	58
Counting occurrences of text .....	58
Upper and lower case -- The CASE cmd .....	59
Searching for "ENTER" -- The "FLOW" CMD .....	59
Marking text – POINTS .....	59
Tabbing .....	60
The Tab keys .....	61
The Grid line .....	61
Setting tabs .....	61
Tab files .....	62
Print-time tabs .....	62
Pitch-dependent tabs .....	64
Why print-time tabs .....	64
Screen print .....	65
Splitting a line .....	65
Blank lines .....	65
Joining lines together .....	66
Reversing two words .....	66
Underline on screen .....	66
Hard spaces .....	67
Graphics and special symbols .....	67
Table of Symbols .....	68
Repeating special characters .....	68
Half-screen scroll up .....	68
Automatic line joining .....	68
Controlling screen size .....	68
Autosave .....	70
Printer selection .....	70
Repeating and displaying last command .....	70
Validating Emphasis Marks .....	71
Commands to control files .....	71
SAVE .....	72
Killing (removing) a file .....	72
Changing the file name .....	72

## CHAPTER 4 -- Continued

Exiting from ALLWRITE .....	73
END command .....	73
Switching files .....	73
NEW command .....	73
Running another program .....	74
OPTIONS command .....	74
Editing 3 files at a time(Mod 4 only) .....	74
Long documents .....	76
Linking files together .....	76
Recommended file sizes .....	77
The LINK command .....	77
Choosing File names .....	78
Imbedded files .....	79
Nested imbeded files .....	79
Example of "Imbed" .....	79
Using several diskettes .....	80
Imbeding part of a file .....	81
Labeled text blocks .....	81
GO control word .....	81
User-definable (Soft) keys .....	83
Soft key files .....	84
Repeating soft keys .....	85
Other considerations .....	86
A soft key to customize the editor .....	86
A soft key for cross-file changes .....	87

CHAPTER 5 - THE TEXT FORMATTING LANGUAGE .....	89
Rules for using Control Words .....	89
Breaks in text .....	90
Rules for Emphasis Marks .....	90
Default formats .....	90
List of Emphasis Marks .....	91
Printer limitations .....	92
Running the Text Formatter .....	92
Run-time options .....	93
Pausing printing .....	93
Exiting from the Formatter .....	93
Formatter exit menu .....	94
Printer status after printing .....	94
Cut sheet (separate pages) .....	94
Mini-Edit .....	95
Organization of this chapter .....	96
Margins .....	97
Letterhead paper .....	98
Hyphenation .....	99
Indentation .....	99
Hanging indents .....	100
Hanging indents with tabs .....	101
Centering .....	102
Blank lines .....	102
Blank screen lines .....	103
Paragraphs .....	103
Numbered paragraphs .....	104

## CHAPTER 5 -- Continued

Control Word Symbol ";" .....	105
Emphasis Mark Symbol "@" .....	105
Line spacing .....	105
Paper size .....	106
Justified and as-is printing .....	107
Margin justification .....	108
Connecting lines together .....	108
Text breaks .....	109
Pagination .....	109
Widows and Orphans .....	109
Avoiding split text .....	109
Turning off page numbering .....	110
Printing selected pages .....	110
Starting page number .....	111
Cut sheets .....	112
Page titles .....	112
Top titles .....	112
Page numbering in titles .....	113
Where a top title prints .....	114
Format of titles .....	114
Emphasis Marks in titles .....	114
Cancelling titles .....	114
Examples of titles .....	114
Bottom titles .....	115
Odd and even page titles .....	115
Pitch (Font) .....	115
PITCH control word .....	116
Micro-space padding .....	118
Daisywheel proportional printing .....	119
Examples for Daisywheels .....	119
Character widths .....	120
Standard Daisywheel tables .....	120
Disk-resident daisywheel tables .....	121
Non-standard print wheels .....	121
Width tables .....	121
Using a Translation Table .....	122
Using a width table .....	112
Wheel changes .....	112
Miscellaneous Control Words .....	124
Supressing features .....	125
Dark printing .....	125
Not underlining blanks .....	126
Comments in text .....	126
Keyboard entry of text .....	126
Stopping for operator intervention .....	127
Carbon copies .....	127
Multiple column printing .....	128
Error control .....	129
Printing imbedded file names .....	129
Form letters .....	129
The ";RD" control word .....	130
Format a name and address file .....	133

## CHAPTER 5 -- Continued

Other Form Letter features .....	136
High volume form letters .....	137
Printing directly from memory .....	137
Mailing labels .....	137
Default labels setup .....	138
Sheets and continuous forms .....	138
Note to NEWSCRIPT users .....	139
Legal documents .....	139
Q&A indicators .....	139
General line numbering .....	140
Creating a Table of Contents .....	141
Suppressing printing of T/C .....	141
Delayed printing of T/C .....	142
Creating an index .....	144
Maximum size of index .....	144
Suppressing an index .....	144
Indexing techniques .....	145
Creating Footnotes .....	146
Specifying footnote formats .....	149
Special considerations .....	150
Character substitution at print time .....	151
Special symbols & Escape Sequences .....	152
Variables and Logic .....	152
Assigning values to variables .....	153
concatenation .....	154
Arithmetic .....	153
Logic (IF) .....	154
Ignoring and labelling text .....	155

CHAPTER 6 -- PRINTER SETUP .....	157
General considerations .....	157
Serial printers .....	157
Anadex .....	158
C.ITOH .....	159
Diablo .....	159
Epson .....	160
Microline .....	161
NEC .....	161

CHAPTER 7 -- ERROR MESSAGES .....	163
Salvaging text from a bad disk .....	163
How messages are numbered .....	163
DOS error messages .....	164
Keyboard driver error messages .....	166
Editor error messages .....	166
Text Formatter error messages .....	168
Text Formatter prompts .....	170

CHAPTER 8 -- SUMMARY OF FEATURES .....	173
Editor .....	173
Cursor controls .....	173
Control keys .....	173
Soft key defaults .....	174
Commands .....	175
Text Formatter .....	175
Emphasis Marks .....	176
Control words alphabetically .....	176
Control words by topic .....	178
CHAPTER 9 -- WRITING WITH A WORD PROCESSOR .....	183



## CHAPTER 1 INTRODUCTION

ALLWRITE is a word processor system for the TRS-80 Models I, III, 4, and work-alike computers such as the LNW-80, and the MAX-80. It has many features to help you create and print a wide variety of documents, ranging from simple letters and form letters to large books, ALLWRITE supports the advanced features of today's "smart" printers, and does so in such a way that you won't need to know how your printer works, but only whether it can do something. This includes right-justified proportional printing on most printers having that capability.

ALLWRITE can process very large documents as well as small ones; it has excellent support for "legal" documents and can print mailing labels, footnotes a table of contents, and an index. Multiple-column layouts can be used on all printers. All of these features, and many others, are standard, not extra cost add-ons.

ALLWRITE is easy to learn because you only need to become familiar with a few features to use it effectively. Later on, when you want to do something more advanced, you'll be able to use either the built-in "HELP" facility, or refer by topic to the appropriate section in this book. As the tutorial in Chapter 3 will show you, the features of ALLWRITE are based on the first one or two letters of common English words: "CE" stands for "Center", "Search and Replace" do just that, and so forth.

ALLWRITE is fast to use, partly because its functions are performed quickly, and partly because it takes an efficient approach to working within the capabilities and limitations of the computer. For example, it can read a 4,000 word (25,000 characters) file from a floppy disk in about six seconds. So, instead of having the computer move text back and forth between memory and disk when you least expect it, you can work on 8 to 10 pages (single-spaced) at a time. To switch to the next, or previous section of a large document takes only two keystrokes, plus the few seconds needed to write out the current text and then read in the next section. All your settings, including "search" arguments and tabs, are remembered when you do this, so you can just continue working from where you left off.

ALLWRITE is SAFE to use. In fact, safety (accidental loss of text) was one of the most important factors in its design and implementation. If you try to delete more than one line of text, you'll be told how many characters will be deleted; unless you confirm the deletion, it won't happen. If your computer "crashes" (stops working for reasons best known to the gremlins that live inside the circuits) but doesn't lose power, you can restart the machine, then tell ALLWRITE to salvage the text that's still in memory. If a disk error causes part of a file to become unreadable, ALLWRITE will salvage the rest of the file, notify you of the error, and mark the lost area. Since large documents are segmented into medium-sized files (4,000 words or so), you can limit the problems caused by defective disks or intermittent recording errors.

If you already know how to use NEWSCRIPT (another word processing system from PROSOFT), you'll find it even easier to fully master ALLWRITE: the Editor is similar, and the text formatting methods are just about identical. The main differences are speed (ALLWRITE is much faster), removal of most limitations (horizontal scrolling and joining short screen lines are automatic now, and extensive tabbing facilities are included), and many new features (most of the ones our NEWSCRIPT customers requested over a three year period).

### **HOW TO USE THIS BOOK**

Most of this book is organized by topics. CHAPTER 2 will help you install ALLWRITE so that you can use it with your computer and printer. Chapters 3,4, and 5 describe all the features of ALLWRITE in a tutorial manner, and when 'text deletion' is explained, all the ways of doing it are shown together. Chapter 8 summarizes the features and gives cross-reference page numbers to the full explanations in the earlier chapters. The other chapters discuss error handling and considerations for certain printers.

If you haven't used NEWSCRIPT or an IBM "mainframe" word processing system before, we suggest you read and use the first three chapters of this book, skipping topics that don't apply to you. They will guide you through the installation of ALLWRITE, and then teach you enough to write, revise, and print a letter. The remaining chapters can be left for later, since they cover the more advanced and specialized capabilities of ALLWRITE. However you

won't always be a beginner, and as your word processing expertise increases, you'll find uses for many of the extensive capabilities we've built into ALLWRITE.

This book has been specially designed for use while you are sitting at your computer: the binding is durable and allows the book to lay flat. The pages turn easily and won't tear out under heavy use. The smaller page size is easier to read and more likely to fit in the limited space most of us seem to have next to our machines.

Many people want books like this one to be very short, and yet cover everything they will ever need to know about the subject. Obviously, those two objectives conflict, so we will offer you a compromise: the book will try to cover "everything you might ever want to know about ALLWRITE," but you won't have to read all of it. There isn't going to be a test on Friday (unless you're using it in a class) and it certainly isn't a novel with a surprise ending. The book is arranged by topics, and there are lots of headings to signal what's coming up next. If a subject isn't relevant to you, just skip it. When you go through the "installation procedure" in the next chapter, you'll see that it covers several computers and several DOS's: just find the sections for your system, and ignore the others. When you use the Tutorial in Chapter 3, you can retry the examples until they make sense, or skip them when they're obvious to you.

If the "tutorial" presentation style is too long-winded for you, the overviews at the beginning of Chapter 3, 4, and 5, plus the Command Summary in Chapter 8, may be more to your taste. The blue "Quick Tutorial" card may also be helpful in getting started, and you can switch back to the book afterwards.

Another approach is to try plunging right in, and to rely on the on-line "HELP" file or the "Cue Card" when you don't know what to do next. Although we don't recommend that approach, we know that some of the people who helped us test the early versions of ALLWRITE learned it that way: they only used the book as a last resort. That didn't seem to slow any of them down, or if it did, they didn't admit it to us.

Now, if you're a NEWSCRIPT "expert", we still recommend that you read these first three chapters! You'll be able to skim a lot of the material since you'll find you know most of it already, but please be aware that some things are different. In particular, ALLWRITE's editor is somewhat different from NEWSCRIPT's, although it is based on many of the same principles. For example, CLEAR "S" is "sentence delete", not "save"; but the "SAVE" command still works exactly the way it did in NEWSCRIPT. If you look over the complete list of control key functions in Chapter 8, we think you'll agree they are even better than the ones in NEWSCRIPT; but you'll have to get used to them.

Finally, if you've been using another word processor, we ask you to approach ALLWRITE as though you don't know anything about word processing. ALLWRITE has its own way of doing things, so please try to withhold judgement until you've gotten used to its methods. In the course of supporting over 10,000 NEWSCRIPT and ALLWRITE users since 1980, we found that people with no word processing experience invariably found it easier to get started than did people who were used to "SCRIPSIT" or "PENCIL". This wasn't because one was better or worse than the other, but because they were different. So, please give it a chance, and you'll find it quick, easy, and logical to learn ALLWRITE.

## **EQUIPMENT REQUIREMENTS**

### Computers:

- TRS-80 Model I, TRS-80 Model III, LNW-80, MAX-80
  - 48K of memory
  - (extra memory on LNW, MAX-80 is not used by ALLWRITE)
  - 2 disk drives or 1 disk drive and a hard drive
  - 64 x 16 screen supported
  - 80 by 24 adaptations NOT supported

- TRS-80 Model 4, 4P, or hardware-compatible equivalent  
64K or 128K of memory  
(ALLWRITE supports the extra memory)  
2 disk drives or 1 disk drive and a hard drive  
80 by 24 screen supported only in native Model 4 mode  
(Model 4 version of ALLWRITE required)  
64 by 16 screen supported only in Model III mode  
(Model III version of ALLWRITE required)
- two disk drives (or one floppy and one hard drive)
- 5-1/4" 35/40 track disk drive to read distribution disk
- ALLWRITE may be used with any floppy or hard drive
- a printer (three can be supported at one time)
- optional RS-232 (serial port) for serial printers (not supported on MAX-80)
- optional, external speaker, connected to tape port
- Model 4 internal speaker in Model 4 mode only

**DISK OPERATING SYSTEMS SUPPORTED**

DOSPLUS 3.3, 3.4, 3.5, 4.0 (Models I, III, LNW)  
DOSPLUS IV (Model 4)  
LDOS 5.1 and up (Models I, III, MAX-80, LNW)  
MULTIDOS (Models I and III, LNW)  
NEWDOS/80 Version 1 (Model I, LNW)  
NEWDOS/80 Version 2, 2.5 (Models I, III, LNW)  
NEWDOS 2.1 (Model I, LNW)  
TDOS 3.4 (Models I, III, LNW)  
TRSDOS 2.3 (Model I)  
TRSDOS 1.2, 1.3 (Model III)  
TRSDOS 6.0 and up (Model 4)

**Software Compatibility**

- compatible with most "high-memory" drivers  
(all those that maintain the "HIMEM" value)
- compatible with LDOS KI/DVR and filters
- integrated with Electric Webster Spelling Checker  
(Electric Webster is sold separately)
- integrated with DOTWRITER graphics text formatter  
(DOTWRITER is sold separately by PROSOFT)
- can use name and address "extract" files created by MARVELIST, which is sold separately by PROSOFT
- includes program to convert NEWSCRIPT files
- reads and writes "ASCII" files

*a few features only work in native Model 4 mode*

**CUSTOMER SUPPORT**

We offer on-going support to our registered customers only. If you can't figure out how to do something with ALLWRITE, please call or write to us; we'll help you if we can. If you find, or suspect, problems ('bugs') with ALLWRITE, we'll try to fix them. There's no particular time limit on this support, as long as you are the original owner of a legitimate copy of ALLWRITE and have not violated the terms of the License Agreement.

If you want to call us, we are open from 9 A.M. to 5 P.M. Pacific Coast time, Monday through Friday, except on holidays. Pacific Coast means that when it's noon in New York, it's 9 A.M. here, and we are just opening. We cannot accept collect calls or return long-distance calls. Our technical service telephone number is shown below, with our address. The toll-free numbers ring at an order-taking service in another city.

If you want to write to us, our address is shown below. Please see "Reporting Problems" later on to find out what materials we may need to help you.

**PROSOFT**  
**Box 560**  
**North Hollywood, CA**  
**(818) 764-3131**

We have found it necessary to set certain limits on our support. Until we receive your registration card, our service will be limited to assistance in installing ALLWRITE, including replacement of defective disks. If the answers to your questions can be found easily in our documentation, we may refer you to a section in this book that will answer your questions, since we expect you to check the book first (that's why it has an index). On the other hand, if you've checked the book, and still can't figure out what to do, don't hesitate to call or write for help: That's why we're here. We can't solve problems with your equipment, DOS, or other software; and in the rare cases where the number of questions from one person or company becomes excessive, we may have to curtail our support.

Finally, if we receive calls or letters from people who have obtained unauthorized copies of ALLWRITE, we will cancel the license on the original copies. It's in your best interest not to give or trade copies of our software, since access to our on-going support and updates is one of the most important benefits available to you. If your friends or business associates want to use ALLWRITE, please suggest they do what you did: buy a copy.

**REPORTING PROBLEMS**

If your computer cannot "read" or "convert" our disk when you receive it, just send the disk back with a note of explanation, and we'll replace it, free. All our disks are verified twice when they are produced, but disks can be damaged in shipping, and some disk drives can't read disks created on other disk drives. This is normal with floppy disks and micro-computers, and we'll take care of it for you immediately.

We've tested ALLWRITE thoroughly, and two dozen of our most careful NEWSCRIPT customers were kind enough to field test it before we released it to the public. So, when we made your disk, we knew of no problems in ALLWRITE. However, if you think something is broken (a "bug") in any of our programs, we want to know about it and try to help you fix the problem.

When you report a problem, please describe it as specifically as possible. If the problem is in printing, then send us a disk containing the file that printed incorrectly, a printed sample of the result, and your original ALLWRITE disk. If the problem is with the Editor, try to describe exactly what happened. ("It crashed" is not too exact.) Remember: before we can fix a problem, we must be able to recreate it.

Updates and corrections to software errors will be offered ONLY to registered licensees of ALLWRITE, so if you haven't sent in that registration card yet, now's the time to do so. Other than replacing defective disks, our customer support group cannot help you until we receive your card.

#### How To Report a Problem

If you think you have found a "bug", call or write to us first; we probably can solve your problem with a simple answer. If you send in a disk for correction, we will need five things (always):

1. your original ALLWRITE disk (with our label);
2. a backup copy of your "working disk";
3. a sample file on disk that re-creates the problem;
4. print samples showing the error (if relevant);
5. a letter explaining the problem and giving your name and address.

We will return the disks with either a correction or an explanation of the error. Please protect the disks with cardboard: they're surprisingly durable, but don't spin too well if creased.

To obtain corrections and updates, you must send us your original ALLWRITE disk (the one with the PROSOFT label on it). Printouts, sample disks, and letters of explanation will be helpful, as explained above. You may want to call us before sending extra materials in, since we may already know about your problem and be able to help you over the phone. When you send disks to us, please protect them with cardboard: they're surprisingly durable, but don't spin too well if creased.

#### SERVICE CHARGES

During the ninety days beginning when you first receive ALLWRITE, we will replace defective disks and attempt to correct reproducible programming errors at no charge. After 90 days, all corrections and updates to disks will be subject to a small service charge (please check with us for prevailing prices if you need an update), and payment must accompany your original ALLWRITE disk. The service charge is for updates only, and does not cover enhanced versions of ALLWRITE, should such versions become available in the future. If you need an update and your copy of ALLWRITE is more than 90 days old, please check with us for current charges.

Enhancements are called "upgrades". They differ from "updates" in that they add worthwhile new capabilities, not just corrections to errors and minor improvements.

#### KEEPING YOU UP-TO-DATE

ALLWRITE already includes almost every useful word processing feature we could fit and make work on the TRS-80, so we won't deluge you with offers to sell you additional features that should have been there in the first place. If we find any serious problems, we will notify you of them, but will not notify you of minor corrections or minor enhancements: I don't like receiving junk mail, and assume you don't either.

If you do need an update, it most likely will be for support of a new printer. You can check with us to see if we've already added the support you need. If not, please send us a copy of the printer's documentation. If it's feasible to add the necessary support, we will do so. The fee for this will be the same as for any other minor update. If you've just bought ALLWRITE, there will be no charge, of course. We cannot promise to support all printers.

## TERMINOLOGY

We'll be using certain words and terms throughout this book, and will try to define them the first time they are used. Each disk you use contains a "Directory," which may be thought of as a list of the things stored on that disk. We will call those things "files," no matter what they contain. A computer really uses only two kinds of files: "programs," which contain commands that tell the computer what to do; and "data," which contain information used by those programs.

The computer is controlled mostly through the programs called the "**Disk Operating System**," or "**DOS**". The DOS runs the computer, lets you store and retrieve files on disk, and contains functions that make it easier for other programs (such as ALLWRITE) to operate and control the computer. The DOS can do other things as well, but we won't be concerned with them here.

ALLWRITE's capabilities are mostly divided into two major areas: "**Text Editing**," which lets you type and revise your letters and papers; and "**Text Formatting**," which lets you define how things should be printed. Text covers anything you type into the EDITOR, from letters to books. After you've finished editing, your document will be printed by the **Formatter**.

The Editor uses a blinking square, called the "**cursor**," to show you where you are in the text. You can move the cursor around by pressing any of the four arrows. You can tell the Editor to do other things by using "**control keys**" that are one-key operations such as "**D**" (delete); or by using "commands" that are two-letter abbreviations of English words. To use a control key, you must press "**CLEAR**" and then one other key; to use a command, you must press "**BREAK**", type in the command, and then press "**ENTER**". Control keys and commands affect what the Editor displays on the screen and stores in the text.

Text Formatting is described through two-letter "**control words**" such as "**CE**" (center), each of which begins with a semicolon; and through special "**Emphasis Marks**" that cause underlining, boldfacing, etc. Control words and emphasis marks determine how things get printed: they control the Text Formatter. As far as the Editor is concerned, they are just part of the text.

Most of this book will teach you how to use these control keys, commands, control words, and emphasis marks to produce completed documents as easily as possible. Although there are hundreds of capabilities in ALLWRITE, only one or two dozen are needed most of the time, and Chapter 3 will explain those common ones.

Some commands are used by themselves, while others can be followed by extra words or numbers that let you give ALLWRITE more detailed instructions. Those extra things are called "**parameters**." For example, the DOS command "**DATE**" lets you enter today's date. The word "**DATE**" is a DOS command, but the date itself is a parameter.

Some commands allow you to use more than one parameter, and when that happens, the parameters must be separated from each other so that ALLWRITE can tell where one ends and the next begins. Sometimes spaces are good enough for this, but sometimes, you will want spaces to be included as parts of the parameters. Commands that allow that sort of thing let you separate the parameters with "**delimiters**," which are characters you choose yourself. What's special about delimiters is that 1) the same symbol must be used as the delimiter throughout the entire command line, even though you can pick a different character the next time you use any command; and 2) the symbol you choose cannot occur in the parameters themselves. Generally, you will want to use something like a slash "/" or period "." as a delimiter, and avoid letters and numbers.

When we ask you to "**RUN ALINSTAL**" (or anything else), we mean you should type that word as a command to DOS, and then press the "**ENTER**" key. "**BASIC**" is not involved. *even though we used the word "run" here. When we show things in quotes, you should NOT type quotes, only what is in them.*

**Notation**

When we show you an ALLWRITE feature in this book; we'll usually do by example:

**;CE 5**

Shows you how to enter 5 lines of text. However, most features are flexible, which means you can use them in fancy ways as well as in simple ones. When the simple forms are used, ALLRIGHT makes assumptions about what is missing. The values that get filled in for you are called "defaults," and were chosen by us because they are the alternatives most likely to fit the situation. The values themselves are called "parameters", and can be numbers, words, letters, or special symbols, depending on the feature involved. If you use ";CE" without a number, only one line will be centered.

Just showing an example or two can't cover all the possibilities of such features, so we'll also use a standard, formal "notation" to describe the possible forms of many features. In this notation, anything that is CAPITALIZED must be typed exactly as shown (the ";CE" in the example above), and anything that is in lower-case must be replaced by a name, number, letter, or word of your own. Things in [square brackets] are optional, and you can omit them if you don't need them, or if the standard "defaults" are satisfactory for you.

For instance:

**;PP [n] [,b]**

";PP" tells ALLWRITE to start a new paragraph. Everything else is in square brackets, which makes them optional. 'n' is an optional parameter used to define the indentation on the first line of a new paragraph; the default for this "5", or one-half inch. 'b' is also an optional parameter. If used, it would have to be separated from the first number by a comma. 'b' stands for the number of blank lines you want to leave between paragraphs; the default for it is "1" blank line.

That's about as abstract as it's going to get; we didn't write this book to get a good grade in advanced symbolic logic, but to communicate our ideas to you, as clearly as we could.



## CHAPTER 2

### INSTALLATION

“Installation” is the process of tailoring ALLWRITE to meet your particular needs. You’ll find this easy to do because we’ve included a working DOS and a self-starting procedure that will make Backup disk for you. If you want to use this DOS, just put the disk in drive zero, have a couple of new, black diskettes available, and press the “RESET” key. You’ll see instructions on the screen, and all you will have to do is put diskettes in the drives when asked to do so. After the disk are made, you’ll be asked to identify your printer and some other preferences. When the procedure is done, you will be able to start using the results immediately.

If you want to use a different DOS, you can copy our files to your own diskettes. We will show you which files you’ll need for this. Then, you will run “**ALINSTAL**” to identify your printer and other preferences. Once it’s done, you’ll be able to start using ALLWRITE.

Although the procedures explain themselves on screen, this chapter will offer you further details and information regarding the steps involved in installing ALLWRITE. Printers that become available and supported by ALLWRITE after this book was printed will be covered in supplements, please contact us as explained in Chapter 1.

To install ALLWRITE, you will need at least two disk drives. If you have a one-drive system, you will need to use someone else’s computer for installation. (Your local Radio Shack may be helpful.) Once ALLWRITE has been installed, a one-drive computer can be used for everyday operations, but some features may not be useable if there is not enough disk space. A one-drive TRS-80 is difficult to use for any purpose, so we recommend using ALLWRITE with at least two drives.

#### **WHAT YOU SHOULD HAVE RECEIVED**

Your ALLWRITE package should have included the materials listed below. If anything is missing, please contact whoever sold you the package.

- This book
- A disk
- A PROSOFT Registration card
- Four small Cue cards with boxes on them
- If needed, some Supplement pages

#### **Outline of Automatic Installation Procedure**

This method can be used on a Model I, III, or 4 if you want to use the DOS we’ve included. Otherwise, you can follow the “Do-it-yourself” procedure, which is shown next:

1. Insert Side 1 of the ALLWRITE disk in drive zero
2. Have at least two new, blank diskettes handy
3. Press the “RESET” button on the computer
4. Follow the instructions and prompts on the screen
5. Answer the Installation questions when they are asked
6. Set switches on your printer, if necessary
7. Place a “soft key” Cue card above your keyboard
8. Start using the resulting everyday “working disk”

**Outline of Do-It-Yourself Instructions Procedure**

1. CONVERT or REPAIR the distribution disk, if necessary
2. COPY the files to your own disk, or BACKUP the disk
3. Run ALINSTAL (the installation program)
4. Make BACKUP copies of the finished disk(s)
5. Set switches on your printer, if necessary
6. Place a "soft key" Cue card above your keyboard
7. Start using the resulting everyday "working disk"

**THE SOFT KEY CUE CARD**

Your ALLWRITE package included four long cards with boxes on them. These are called "Soft Key cue cards." All of them are similar, and divided into three areas. You only need to use one of them: the others are supplied as spares and to let you fill in your own definitions later on. The cards fit in the slot behind the upper-most row of keys.

The bottom third of two of these cards is filled in, while the bottom of the other two is left black for your use in the future. Take one of the filled-in cards and fold it down on the dotted line (the cards are scored so that they will fold properly). Then, just drop the narrow "leg" into the slot just behind the keys, and the card is ready for use. You may want to fold it again, upwards this time, along one of the other scored lines. When in place on a Model III or 4, it should cover the bottom of the screen, but should not block the screen's actual display area. The words "New Paragraph" should be directly above the "1" key. If you have an LNW, you may want to tape the card in place; on the TRS-80's, the card fits exactly, and will not move around.

**What The Card Does**

This Cue card will make it even easier for you to use ALLWRITE productively; it shows the settings of the "Soft Keys", which let you perform many of the most common word processing operations with a single keystroke. Later on, we'll show you how to define the unused keys, and you can use the black cue cards if you want to re-define everything. You have our permission to make additional photocopies of the cards for future use. The upper portion of the Cue card lists all the screen functions of ALLWRITE, so without taking any space on the screen itself, you will always have the equivalent of three full screens of "HELP" summary information right in front of you. When you need further explanations, you can use the on-line "HELP" feature by pressing CLEAR and then ZERO.

**BACKUP COPIES OF DISKS AND FILES**

As you'll see throughout this book, we urge you to keep backup copies of all programs and data files that are of value to you. Diskettes are very cheap in comparison to the time you will lose if you have to try to re-create the contents of a diskette that becomes unreadable, or re-type a file you deleted by mistake. One good method for keeping backups is to use three disks: the one you're working with right now, the most recent backup, and the previous backup. When you're ready to make the next backup, you can re-use that "previous" disk, so if anything goes wrong while you're making the backup, you'll still have the "recent" copy. If you're the cautious type (as we are), you may decide to rotate several disks this way, instead of just three.

You should NEVER, ever, use an original distribution diskette for everyday use. Our disks will back themselves up and then customize the backups to meet your requirements. Then, you should make several extra backup copies of the results, and also some backups of the original disk. After you've done all that, put the original away in a safe place: you will need it to obtain upgrades or enhancements from us.

**FORMAT AND CONTENTS OF THE DISTRIBUTION DISK**

ALLWRITE is distributed in different formats to match the computers it supports. The same file names are on all the disks, but their content varies according to the computer for which they are intended. We've taken care of this so you want have to thing about it: if you use the disk intended for your computer, everything you need will be on it.

We use "floppy" disk to distribute many of our programs. A floppy disk has information on both sides of it, but only one side can be used at a time. The "front" or first side, is smooth, while the "back": has the folded seams of the black jacket. To use the second side, just turn the disk over; a two-sided drive cannot read both sides at once.

Except for the side intended for use with Model III TRSDOS, all these disks contain a pre-configured working DOS, with unneeded files removed to make space for ALLWRITE. To start the disk, just insert side 1 (the normal side) into drive 0 and "boot" the computer. If side 2 is needed, the procedure will ask you to turn the disk over. We've paid a license fee to the manufacturers of each DOS, so your copy is legitimate and intended for use with ALLWRITE. We've done this to help you get started as quickly and easily as possible. We have not provided documentation for any of the DOS's: if you already have a normal copy of the same DOS, you also have the documentation; if not, please copy ALLWRITE to your own DOS and use it there.

**Model I**

Tiny DOSPLUS is on both sides of the disk. The one that says "ALWRITE", and on the only side of the supplement disk. The primary disk is in 40-track, single density format. If your disk drives only have 35 tracks, just return the disk to us with a note, and we'll send you a 35-track replacement. Most disk drives have 40-track capability, even though TRSDOS doesn't use the extra five tracks. So, unless you know your drives won't handle 40 tracks, try our disk to see if it works. You may be pleasantly surprised. The TDOS we've included is faster and more reliable than TRSDOS. If you want to use this TDOS, just put Side One of our disk in drive zero, press the RESET key, and follow the instructions on the screen.

If you have a MAX-80, the disk must go in drive 1, with LDOS in drive 0. Remove our write protect tabs and then use the LDOS command, "REPAIR :1 (ALIEN)" to make the disk readable.

If you want to use your own DOS, make BACKUP copies of both sides of our disk, then see the instructions later on for details of which files to copy.

**Model III**

Side One contains a working TDOS (Tiny DOSPLUS) and all of ALLWRITE. If you want to use TDOS, which is much faster and more reliable than TRSDOS, just put side 1 in drive 0 and press the orange RESET button; the automatic procedures will take over and guide you along. If you want to use LDOS, MULTIDOS, or full DOSPLUS, put Side 1 in drive 1, put your DOS in drive zero, make a BACKUP of our disk, then see "BUILDING A WORKING DISK" later on to find out which files to copy.

If you want to use TRSDOS or NEWDOS/80, use Side TWO of our disk: it is a TRSDOS 1.3 data disk (no DOS). See instructions later on to find out which files to copy.

**Model 4**

Side One contains a licensed copy of TRSDOS 6.x (6.1, 6.2, etc.), a self-starting installation procedure, and some of the ALLWRITE programs. Side Two contains a licensed copy of DOSPLUS IV, a continuation of the installation procedure, and the rest of ALLWRITE. Always start with Side One, even if you have your own

DOSPLUS IV. To make room for ALLWRITE, we have removed BASIC and several other unneeded files from both sides. Put Side One of the disk in drive zero, press the RESET button, and then follow the instructions on the screen. The Model 4 version does not have, or need, a supplement disk.

The disks contain these files:

1. AL/CMD	text editor
2. ALF/CMD	text formatter
3. AL/DEF	stores installation specifications
4. ALF/DEF	sets up for special documents
5. ALK/CMD	keyboard support
6. ALKOFF/CMD	disconnects "ALK", Optional
7. HELP/AL	on-line quick-reference file, Optional
8. SAMPLE/LET	used to checkout and learn ALLWRITE
9. NS2AL/CMD	converts NEWSCRIPT files to ALLWRITE
10. DIRLIST/BAS	companion program to "NS2AL/CMD"
11. ALINSTAL/CMD	the installation procedure
12. ALINDEX/CMD	creates sorted index
13. LEGAL/EX	sample file of a legal style document
14. FORMLET/EX	sample file for printing form letters
15. NAMADDR/EX	sample file for Form Letters and Labels
16. PRTABLES/PRT	used by installation procedure only
17. xxx/TAB	tables for daisywheels and thimbles (there are several of these)
18. ALINSTAL/BLD,	this is the automatic self-install
JCL /TXT	procedure and its only needed once

To use ALLWRITE, the first three files must be available. To use certain text formatting features, the fourth file is also needed. Unless you are using LDOS with "KI/DVR or Model 4 TRSDOS 6.X, you should also use the 5th file. If your DOS does not have "type-ahead" and good "debounce", you will also need the fifth file. LDOS and TRSDOS 6.x have very good type-ahead, and if you're using either of these DOS's, you should use their type-ahead, not ours. While learning to use ALLWRITE, the HELP file will be useful, but if it takes too much room, you can keep it on a separate disk.

### **COMPUTER-DEPENDANT CONSIDERATIONS**

Before installing ALLWRITE, you should be aware of the following:

1. ALLWRITE allows screen "lines" of up to 255 characters. If the actual width of the screen on your computer is narrower than the size you select, the text will shift to the left ("scrolling") as you type. This means that actual screen size (80 on a Model 4, or 64 on anything else, even if you have an LNW or MAX-80), isn't a limitation when editing text. The Installation procedure will ask you to set both a "maximum" value (the widest you will ever need), and a "default" value (the normal width you expect to use most of the time). We recommend that you set the "default" width to match the actual screen width. The "maximum" width can be bigger, but you shouldn't make it any bigger than necessary, because the space for the screen is taken away from the space used to edit text. "80" is probably a good maximum value unless you expect to work with wide tables. Since ALLWRITE can print lines that are much wider than the screen, you don't have to set the screen width to match the widest report you'll ever print.
2. Model 4 TRSDOS 6.x has its own built-in type-ahead support, so our "ALK/CMD" must not be used with it. You can increase the speed with which the cursor moves by using these DOS commands:

**SETKI (RATE=1)  
SYSGEN (YES)..**

We did this for you in the TRSDOS we supplied. If you're using DOSPLUS 4, you should use "ALK/CMD".

3. If you have a 128K Model 4, ALLWRITE's editor can use that extra memory. However, if you've told the DOS to put a spooler or MEMDISK there, the space won't be available to ALLWRITE. Also, if you set your maximum screen width much over 125, the editor may not be able to use any of the extra memory, because of the way that the editor divides available space. If this happens and you try to use the unavailable space, you will get a warning message, but no harm will be done. If you don't need to edit several files at a time, you can use the extra memory as a MEMDISK, and copy ALLWRITE's 4 primary files into it: This will speed up the transitions between editing and printing.
4. We do not recommend using the extra memory as a spooler with ALLWRITE, because it may slow things down and, if you're doing proportional printing, will have only 1/3 to 1/4 the capacity you might expect. We do recommend using the extra memory as a memdisk, except when you want to edit a couple of files at once. Whether you put AL/CMD and ALF/CMD there, or portions of Electric Webster, DOTWRITER, or DOS itself in a memdisk is up to you. Don't put your text files in MEMDISK, because you may forget to copy them to a real disk!

### **MAKING A BACKUP OF THE DISTRIBUTION DISK**

"BACKUP" is the process of making a duplicate of a diskette. It gives you some protection against damage to the original, and should be considered an essential part of all the work you ever do with a computer. After you install ALLWRITE, you should make at least one backup copy of the distribution disk we sent you. In general, whenever you receive (or create) an important disk, the first thing you should do is make a copy of it, using the standard procedures for your DOS, and put the master away in a safe place.

If you received a model I disk for your model III, you will have to "CONVERT" it with TRSDOS, DOSPLUS, and MULTIDOS; "REPAIR" it with LDOS; or "WRDIRP 1" it with NEWDOS/80.

### **BUILDING A WORKING DISK**

A "working disk" will be the everyday copy of ALLWRITE used whenever you want to do word processing. Depending on the quantity and capacities of your disk drives, you may want to keep ALLWRITE on the same disk as your DOS, or on a "data disk" (if you have at least three drives), or on a hard disk, if you have one. We will assume you want to keep ALLWRITE on a DOS disk, but if you decide to put it someplace else, the same files and procedures should be used.

The automatic installation procedure on the distribution disk will create a working disk for you, with all the files you need for everyday use. If you use that procedure, you can skip the rest of this section, and pick up again at "Menus and Installation". If you decide to create your own working disk, you may have to "KILL" or "REMOVE" some DOS files to make room for ALLWRITE. Then, you will need to copy some of the ALLWRITE files to your disk.

One problem in using floppy disks is their limited capacity. This problem can be annoying if you have 35 or 40-track single-density drives, but doesn't even exist if you have 80-track or hard drives. If you have a space problem, you should plan to copy only the minimum number of ALLWRITE files to the working disk, and keep the others on backup disks for occasional use.

These files **MUST** be on the working disk:

**AL/CMD ALF/CMD AL/DEF**

If you've decided to use our keyboard support (keyboard "driver"), then this file must be on the working disk also:

**ALK/CMD**

If you have room, or extra disk drives, also copy "HELP/AL". If it isn't on line, you won't be able to use our "HELP" key.

If you plan to use any of these features: Mailing Labels, Legal Documents, Table of Contents, or index; then this file should be on the working disk also (it's on Side 2 of the Model I disk):

**ALF/DEF**

If the above files are not yet on your working disk, use the "COPY" command to move them there now.

Example: **COPY AL/CMD:1 AL/CMD:0**

**TAILORING ALLWRITE TO YOUR REQUIREMENTS**

If you're using the self-starting procedure on the distribution disk, then let it run, and when it asks "Do you want instructions?", skip ahead in this book to the next topic, Menus and Instructions." If you want to use your own DOS, you can prepare a "working disk" for everyday use by copying files from our disk to a BACKUP of your DOS disk. You will need to copy the following files:

**DRIVE 0**

A "DOS" disk. It must contain these files plus two free granules for each of your printers.

**AL/CMD      ALF/CMD      AL/DEF**

ALF/DEF (for legal, labels, Table of Contents, Index)  
ALK/CMD (if you want to use our keyboard support)  
HELP/AL (if you have room for it)

**Any available drive**

ALINSTAL/CMD  
PRTABLES/PRT

When all disks are ready, type this command from DOS (not BASIC):

**ALINSTAL** (and press ENTER)

(The automatic procedure does this for you after creating the necessary Backup / working disks.)

**Menus and Instructions**

The installation program contains detailed explanation and "menus" (lists of possible answers) to guide you along. The material below is supplementary to the installation itself.

When ALINSTAL begins, it will ask you whether or not you want instructions. Just press "ENTER" to get them. If you've run installation several times, and no longer need instructions, reply "NO" instead. It's quicker if you skip the instructions, but if you're using the program for the first time, the explanations will be helpful.

Each time you are asked a question, you can reply with one of the choices given, or press the BREAK key to quit and get back to DOS. Sometimes, you can press the CLEAR key to repeat the last few questions of a series. Usually, after typing an answer, it will be necessary to press the ENTER key to have that answer accepted; until you press ENTER, you can use the left arrow to back up over a mistake.

### Printer Selection

If you ask for instructions, press ENTER each time you've finished reading what's on the screen. After the instructions have been displayed, you will be asked to select your first (or only) printer. The first time you install ALLWRITE, you must select at least one printer. When you re-run ALINSTAL later on to change some other values, you can skip the printer step if you don't want to change any of your earlier answers. This makes re-installation very quick.

Ten printers at a time are shown, and the list is in alphabetical order. If you're printer isn't in the first part of the list, just press ENTER to keep going. When the end of the list is reached, it will go back to the beginning. Radio Shack printers are listed under the "R's", not under "DMP". We suggest you review the entire list (over 70 categories when this book was sent to press) before making any selections.

There are variations on some printers, so be sure to pick the exact one you have. For instance, there are 3 versions of the Epson MX-80, and two versions of the Diablo 630. To select a printer, type the number next to it, and press ENTER. If your printer is not on the list, and you don't have any clues as to which supported printer comes closest to your own, the 2 best choices to try are "DIABLO", "TELETYPE". After completing installation, try printing the "SAMPLE/LET" we've included; if it doesn't print properly, try another printer. If you don't find a good match, see chapter 1 for instructions on what information to send us. QUME and many other daisywheels should be installed as DIABLO's.

Note: ALLWRITE's "STATUS" screen displays the name of the currently selected printer. And so does the Text Formatter (ALF) when it starts. The name may differ from what you've selected because ALLWRITE groups certain functionally similar printers. If everything prints properly, do not be concerned with the name difference.

If you do select several printers, you can switch from one to another with the Editor's "PRINTER" command, which is explained later in the book.

### Printer Attachment

After you select your 1st printer, you'll be asked how it is attached to the computer. Most printers are connected to the "parallel printer port" and some are connected to the "RS-232 Serial" port. ALLWRITE supports both types of connection. If you don't want ALLWRITE to control the printer directly, but to work through your DOS instead, pick the 3rd choice on the "Attachment Menu." The 3rd choice is required on the LOBO MAX-80, but is not recommended in most other cases. Most DOS's cannot control a printer as well as ALLWRITE can, so you may lose some features if you make this choice, and printing will certainly be much slower than otherwise.

### Serial (RS-232) Printers

If you select "parallel" or "user-defined" attachment, the rest of this section does not apply. If you select "serial", you will be asked other questions about the "BAUD RATE" (pick the fastest speed your printer can accept), "PARITY", "WORD LENGTH", and "STOP BITS" (for each of these, pick the settings you're using in your printer). Finally, you'll be asked about the "handshake" method to be used. ALLWRITE offers 3 kinds of support in this area. If your printer recognizes the "ETX/ACK" protocol, pick that when asked; if it uses "DC1/DC3" (also called "X-ON/X-OFF"), pick that. If it offers both ETX and DC1, we recommend DC1/DC3. If your printer recognizes neither of these, then you can match the speed of the printer to the speed of the computer by choosing "padding." "Padding" is a delaying technique, and is needed only if the printer runs a little slower than the BAUD rate. For example, if your serial

printer runs at 30 characters per second (cps) and you set the baud rate at "300", no padding will be needed unless a series of page ejects occurs. If your serial printer runs at 45 cps and you set the baud rate at "300", the printer will run no faster than 30 cps (or even slower if proportional printing is involved). However, if you run it at 1200 baud, you should set the padding at "3" or "4" to give the printer time to catch up.

Remember, "padding" should be used only if neither ETX/ACK nor DC1/DC3 works on your printer. DIABLO's use ETX/ACK, and SPINWRITERS use DC1/DC3. If you aren't sure about this, check your printer manual.

### **Printing with Spoolers**

ALLWRITE's Text Formatter (ALF) has its own built-in spooler (print buffer), and we recommend you use it if possible. If your DOS has a "spooler" capability, and you want to use it, you must select "User Defined printer driver", which is the 3rd choice. However, you should be aware of some drawbacks to using a software spooler with a word processor. 1) it may actually be slower, since some spoolers need to access a disk to read and write the text. 2) some software spoolers modify or discard control characters before sending them to the printer, and that will ruin the print job. 3; when proportional printing is used, 2 or 3 "control characters" must be added to each character that gets printed. This means a 1 page letter, containing 2,000 printed characters, may need up to 8,000 characters of spool space. If you're printing in "monospace" (10, 12, or 16 pitch), this won't happen, but a 30 page proportionally spaced paper could require up to 300,000 characters of spooler disk space, even though it only needed 75,000 characters of space to be stored in its normal, written form.

Some DOS's modify or discard control characters before they are sent to the printer. This will ruin the print job if you select "User Defined Printer Driver," but it won't happen if you use our standard parallel or serial support. You can use the DOS driver in DOSPLUS if you hold down SHIFT UP ARROW when you press RESET. You can tell the LDOS "PR/FLT" to translate everything to itself, including binary zeros and Form Feeds.

If you're using a hardware spooler (an external print buffer), you should use ALLWRITE's parallel or serial support, not the "user driver." Remember that proportional spacing will generate a lot of control characters, so a 64K print buffer will only store about 10 pages at a time, not the 25-30 pages it can store in 10-pitch. In serial mode, most print buffers will not be able to support the "ETX/ACK" protocol, since they won't be able to distinguish between a real "ETX" from ALLWRITE and a printer control command that happens to contain the binary number that stands for an "ETX" ("3").

Whether you use a hardware spooler or a software spooler, it must not modify any characters in any way; if it does, the printer will receive incorrect instructions. The solution is to either disable the modification function, or to not use that particular spooler. This is not due to a problem with ALLWRITE, but to a problem with the hardware spooler.

### **Cut (Single) Sheets**

Most printers use "continuous form" paper, the kind with holes down the sides. If your printer generally uses this kind of paper, just accept ALLWRITE's default. If you use typing paper (single, or cut sheets, rather than continuous forms) and need to have ALLWRITE stop at the end of each page until you have inserted the next page, specify that when asked.

### **Line Feeds**

If your printer does not advance to the next line automatically on receiving a carriage return, tell ALLWRITE to add a line feed for you. The default is to not add a line feed. Some printers cannot fully utilize their high quality modes unless automatic line feeding is turned off.

### MULTIPLE PRINTERS

After you've answered the above questions, you'll be shown the printer menu again. Reply with an (\*) if you have only one printer; otherwise, select your 2nd printer by number, just as you selected the 1st printer. All the same questions will be asked, since ALLWRITE treats each printer completely separately.

If you describe a 2nd printer, ALLWRITE will give you the printer menu a 3rd (and final) time. If you have 2 printers, reply with an (\*) to show you're done. Otherwise, select and describe your third printer. If you have more than 3 printers connected to the same computer, you will need to maintain multiple copies of "AL/DEF", and run the installation program more than once to create these multiple copies.

### KEYBOARD AND SCREEN PREFERENCES

The next series of questions lets you adjust the shape and blink rate of the "cursor" (the line or square that shows you where you are on the screen). You can turn off blinking entirely, but we don't advise it, since you won't be able to see what character is "under" the cursor. Since the original and current values are both shown, you can experiment with the blink rate and re-install until you are satisfied.

When you're entering or revising text with ALLWRITE, the normal cursor shape will be a blinking square. You can pick a different shape, such as an underline character. The shape must be specified as a number between 33 and 255. "95" is the underline character, and is the next most popular shape after the small square. If you just press ENTER in reply to this question, the previous shape will continue to be used.

Pressing ENTER at the end of a screen line marks a breaking point in the text (end of paragraph, for instance). This is stored in the text as a "Carriage Return" (ASCII 13), but to let you know where you've set these breaks, a special symbol is displayed on the screen. By default, we use a graphic that resembles a backward "L" or the "Return" key on many typewriters, but you can change the symbol if you want to. If so, we recommend you choose a graphic character (between 129 and 191), since the character you choose cannot be used as normal text at the end of a line. (Therefore, the period "." would be a bad choice.) You can make this character invisible by using the value "128", but we don't advise it, since you won't be able to see where your paragraphs end, and may erase these markers without knowing it. If you find the backward "L" too large, try "136", which is a small square.

A slowly blinking cursor lets you see what character is "under" the cursor, and also attracts your attention to where the cursor is positioned on the screen. Some people like a non-blinking cursor, and others like one that blinks so fast that it appears transparent. You will be asked to specify how long the cursor should be "ON", and then how long it should be "OFF". We suggest you just press ENTER to answer both these questions, and then see how you like the speeds we've chosen. If you want to change them later on, you can do so easily by re-running one portion of the installation procedure.

As we mentioned earlier, ALLWRITE lets you type lines that are much wider than the actual screen of your computer. You will be asked what the "MAXIMUM" screen width should be, and we recommend "80", unless you expect to work with wide tables from time to time. The smallest value you can use is the true screen width (64 or 80). You also will be asked what the "NORMAL" width should be whenever text editing begins. The answer cannot be larger than the "maximum", and probably should be the actual width of your screen (64 or 80). You'll be able to adjust the current screen width as often as you wish, at any time, while editing text.

### Autosave

When entering and revising text, it's easy to lose track of time, and to forget the importance of saving your work to disk every few minutes. Why is this important (you may ask, if you've never learned the answer the hard way)? Because things can go wrong, and, as "Murphy's Law" states, "usually at the worst possible time." There could be a power failure. A refrigerator could turn on, lowering the electrical power in your house just enough to affect your

computer. Someone might turn the computer off by mistake, without first saving the last hour of work. And, incredible as it may seem, there might even be something wrong with the software (we don't think there's anything wrong, but since we are careful to save text frequently, the software has nothing to gain by failing on us).

The point is, whenever you do anything with a computer, you **MUST** keep extra copies of your work, your files, and your programs. Unfortunately, all of us tend to forget about this when in a hurry. That's why ALLWRITE's editor has a feature called "AUTOSAVE". It will count the number of lines you've typed, the number of changes you've made, and when the total reaches a pre-determined limit, ALLWRITE will automatically save everything in memory onto a disk. The screen will go blank for a moment, and you'll be shown a message telling you the text is being saved (which only takes a few seconds). Then, the text will re-appear, and you'll be able to continue right where you left off.

The "default" AUTOSAVE value is very large: 65535. For all intents and purposes, this means the feature is turned off. You can set AUTOSAVE at any time while editing, but you can also set it during installation (and still change it while editing). If you decide to use this feature, we suggest a setting of "50" or so; experience will tell you whether that's right for you or not. The number to use depends on how much you enjoy re-typing your work after something goes wrong.

### Verifying Disk Writes

For safety, whenever anything is written to disk, DOS should be allowed to immediately check what's been written and verify that it matches what's still in memory. The "VERIFY" command in DOS controls this, but ALLWRITE can tell DOS to verify all disk writing even if "VERIFY" is "OFF." The tradeoffs in deciding whether or not to use "VERIFY" are simple. Writing with verification takes 3 to 4 times longer than writing without it; but if something goes wrong while a file is being written, and verification is not in use, part of that file may be lost, and you won't know about it until the next time you try to read that file. If the file had been verified, you would have been told of the error in time to save the text to another disk.

We only run our computers with "VERIFY ON", and urge you to do so also. However, if you have unbounded confidence in your equipment and diskettes, you can tell ALLWRITE not to force verification when files are written to disk. In that case, the setting of the DOS "VERIFY" command will prevail.

### Audible Click

The final question in this portion of installation lets you set an audible "click" each time a key is depressed when editing text. To hear the click, you will have to connect an external speaker (sold by Radio Shack) to the tape port of the computer. You'll need a connecting cable to do this. ALLWRITE doesn't use the built in speaker of the Model 4, and if you have that computer, you won't want to use our "click" anyway: the one supplied with the operating system should be used instead. Some people like to hear a "click" to confirm that they've really pressed the key, while other people dislike the constant chatter. The default on this question is "no click".

### KEYBOARD DRIVER SUPPORT

If you are using Model 4 TRSDOS 6.x, LDOS with KI/DVR, reply "NO" when asked about our keyboard driver. If you are using any other DOS, you should reply "Y" and always use "ALK" to start ALLWRITE. (The question has no default answer). If you don't want to use our "ALK/CMD" for type-ahead and debounce, or if you're satisfied with values you set during a previous installation, just reply "NO" when this topic comes up during installation; otherwise, reply "YES" (there's no default). You'll be asked only two questions: how fast should the "repeat speed" be, and how much "debounce" is needed to fix up your keyboard. The normal TRS-80 repeats at about 13 characters per second, which most people find to be much too slow. ALLWRITE's default is "30", but you can pick any value between 10 and 100. If you have a "speed up" kit on your computer, take it into account when selecting this speed.

Early TRS-80 Model I's, late Model III's, some Model 4's and many LNW-80's have keyboards with a "bounce" problem. "Bounce" happens when you press a key once, briefly, and then it shows up on the screen twice (or more). Most of the DOS's try to fix this, but some keyboards almost defy a solution. ALLWRITE's "ALK/CMD" has excellent debounce ... good enough to cure even one of our own machines, which has the worst bounce problem in the country. (ALK was written by the #%^2\*# person who has to use that particular computer). If you don't think you have a bouncy keyboard, just set the "debounce" to "1". A value of "3" cures the average keyboard, and "5" cures most of the rest. Please be aware that a large "debounce" value will slow down the "repeat speed" and can cause loss of keystrokes.

Some DOS's do not recognize lower-case commands, and will give an error message (Error 19) when you use lower-case. This problem can happen after you've been using ALLWRITE in lower-case, and then exit to DOS to do something else. The solution, of course, is to press SHIFT ZERO to get back to upper case.

"ALK" sets lower-case automatically; If you find the Editor running in upper-case, you probably started with "AL" instead of "ALK". (This does not apply to operation under Model 4 TRSDOS.)

### **OPERATING SYSTEM**

The next question will ask you to specify the DOS you will be using. If you want to use ALLWRITE with a different DOS later on, you must re-install (run ALINSTAL again) and re-specify the operating system. If you forgot to do this, ALK may not work, and the ends of files may be lost or filled with incorrect text.

Special note to MULTIDOS users: our DIRECTORY commands won't work with some MULTIDOS disks because their directories are protected in ways we cannot bypass. You can still use MULTIDOS's directory commands before you activate ALLWRITE, or contact CEC to order a more current release of MULTIDOS.

### **FINISHING INSTALLATION**

The final question asks you whether to save your answers, start over, or just quit. There's no default, so if you're satisfied with your replies, answer "YES" and press ENTER. The disks will run for a few seconds, and if everything goes well, you'll be given a "success" type message. If something goes wrong (a file isn't available, a disk error occurs, or the disk is write-protected), you'll be given an error message and a chance to fix the problem and try again.

### **Daisywheel Printers**

Please read this section if your printer has interchangeable wheels or thimbles, is capable of "proportional" printing, and is not a Radio Shack printer. The ALLWRITE distribution disk (or supplement disk) has several small files whose "extensions" (the part after the "/") are "TAB". These are tables, used by ALLWRITE to describe the width and sequence of characters on the wheels and thimbles used by printers such as DIABLO, QUME, C.ITOH, and SPINWRITER (and other printers that are compatible with these). Tables for the most common wheels and thimbles are built into the printer tables themselves, and identified in the "PITCH" topic of chapter 5. Please refer to that topic if you have a Spinwriter, Juki, or Diablo-compatible printer. If you need any of the disk tables for the other wheels or thimbles, you should COPY the appropriate file(s) to your working disk after completing the installation program.

### FINISHING TOUCHES

If your "working disk" contains DOS as well as ALLWRITE, you can set the DOS command, "AUTO" so that ALLWRITE will start automatically whenever you "boot" the computer. Depending on whether or not you are using our "ALK" keyboard driver, the command to use is:

**AUTO ALK**  
**AUTO AL**

and then press ENTER.

### Printer Description Tables

The installation procedure creates a small file describing the characteristics of your printer. If you've selected more than one printer, there will be a separate file for each of them. The name of the file will be similar to the name of your printer, and the "extension" (the part after the "/") will be "TAB", for "table." The printer table(s) will be placed on whichever diskette your DOS chooses, but you will want it (or them) to be stored on your "working disk" for daily use. Please check the directories of your diskettes to make sure these files are where they should be, and COPY them to the working disk if necessary. ALLWRITE cannot control your printer(s) properly unless it finds the correct table file, so be sure not to skip this step!

You can edit the printer "xxx/DEF" files and "ALF/DEF" but you cannot edit the printer "xxx/TAB" or "AL/TAB." If you try to do so anyway, and save them back to disk, they won't work afterwards.

### AFTER YOU'VE FINISHED INSTALLATION

When you reach this point, your "working disk" should be ready for use. Before starting to use it, please put a label on it and make 1 or 2 BACKUP copies of it. Also at this time, you should put the original ALLWRITE disk away in a safe place. You may need it if you forgot to copy something, and you definitely will need to send it to us if you ever want an update, correction, or later version of ALLWRITE. The sorry state of software piracy being what it is, that disk is the best proof you can have of your legitimate ownership.

If you have at least 2 disk drives, you should put a write-protect tab over the square notch on the side of the working diskette, (A write-protect tab is one of those sticky silver, white or black things that come with disks.) That way, you won't accidentally write text files onto that disk.

### RE-RUNNING INSTALLATION

You can re-run the installation procedure at any time by typing "ALINSTAL" when "ALINSTAL/CMD" is on line. If you want to pick different printers, "PRTABLES/PRT" must also be on-line. If you want to change repeat speed or debounce, "ALK/CMD" must also be on-line, on a disk that is not write-protected.

### One-Drive Systems

If you have only one drive, you can put the ALLWRITE programs and support files on one DOS disk, and keep several other DOS disks for data. Remove (kill) as many unnecessary files as possible from those "DOS/data" disks to give yourself some room. When ALLWRITE asks you for the file to be edited, just switch disks. That method works pretty well, for those of you with only one drive probably know already.

**DATA DISKS (2 or more drives)**

Your text files should be kept on "data disks", the kind that go in drive 1 of the computer. These must be "formatted" first, and the "FORMAT" command of your operating system is used for that purpose. It's a very good idea to keep a few formatted disks around, because when the one you're working with fills up, and you have unsaved text in memory, it will be too late to format a disk on the spot. Too many people let this happen (although seldom more than once), but you will save yourself a lot of grief by taking a few minutes right now to format, label, and set aside a couple of extra data disks.

By the way, if you have placed a write protect tab on the working disk, it probably will have enough room to get you out of this situation (once). That's another reason for protecting it.

**SWITCH SETTINGS ON YOUR PRINTER**

This completes the installation procedure for ALLWRITE. However, some printers use internal "DIP" switches that control such things as line feeds and buffer size, and these switches often are set incorrectly for proper use with a TRS-80 or ALLWRITE. If you've set those switches so that they work with NEWSCRIPT, they will still work here; otherwise, it's possible that your printer has just "happened" to work with your computer, simply because its special features haven't been exercised until now.

Chapter 6 contains information about printer switch settings, and we suggest you look up your printer there if you have problems printing documents with ALLWRITE. New printers will be announced after this book has been printed, so even if your printer isn't listed, its predecessor may be there. Also, of course, your printer manual should explain what the switches do. We've only listed printers for which switch settings have caused confusion in the past.

The remaining topics in this chapter will be of interest mostly to people who use NEWSCRIPT, ELECTRIC WEBSTER, or DOTWRITER.

**CONVERTING NEWSCRIPT FILES TO ALLWRITE**

This section will be useful if you have files in NEWSCRIPT format and want to use them with ALLWRITE (NEWSCRIPT is another Word Processor available from PROSOFT). We've included 2 conversion programs for you: They work only in Model I or III mode, and one of them uses BASIC, so you may have to use your old NEWSCRIPT/TDOS disk to run it.

ALLWRITE's text formatting language is almost identical to the one used in NEWSCRIPT, ("SCRIPT control words"), so you will need to learn very little more to use it. A few changes were made so that ALLWRITE's language would be easier and more logical, several new control words have been added to support new features, the "boldface escape sequence" has been changed to allow nine degrees of darkness, and the notation for sub/super-scripts has been reversed in response to requests from many of you.

To let you use your existing NEWSCRIPT files with ALLWRITE, two conversion programs have been supplied. They will convert your NEWSCRIPT files quickly and automatically, so in most cases you won't need to do any further editing to use the results. The programs are:

**DIRLIST/BAS  
NS2AL/CMD**

These programs cannot be used in native Model 4 mode. They are the only components of ALLWRITE with that restriction, but if you have been using NEWSCRIPT, you've been doing so in Model III mode anyway. Once your files have been converted, you can just read the resulting disk(s) in native Model 4 mode with ALLWRITE.

Converted files usually are a little bigger than the originals, so make sure there's some room on your disks before starting.

### **DIRLIST/BAS**

This program creates a list of files so that "NS2AL/CMD" can convert an entire disk automatically. "DIRLIST/BAS" reads the directory of a Model I or III disk, and writes an "ASCII" file, containing all the filenames it finds, back onto the same disk. You can edit that list with NEWSRIPT or ALLWRITE to remove any files that should not be converted. Use of this program is optional, since "NS2AL/CMD" can convert individual files if you don't want entire disks converted automatically, or if DIRLIST/BAS doesn't work with your operating system. Its in BASIC to let some of you change it to work with your DOS, if you are so inclined. The program should work as-is with DOSPLUS, LDOS, and NEWDOS/80; it will not work with TRSDOS and probably will not work with MULTIDOS.

Before using DIRLIST/BAS, make a BACKUP copy of each NEWSRIPT text disk you wish to convert. This is a safety measure: when NS2AL/CMD is used in automatic mode, it replaces the files in place as it converts them.

To use DIRLIST/BAS, start BASIC with at least 2 files, and then RUN "DIRLIST/BAS". It will ask you for the drive number containing the disk to be listed. Put a BACKUP disk containing NEWSRIPT text files in a drive (probably drive 1), and answer the question. The program will read the disk directory (if possible), then write the names it finds into a file it will create on the same disk.

After the program finishes, you can re-run it if you have other disks to process. You might want to process just one disk with NS2AL before converting the other disks, to satisfy yourself that everything works properly. You also should put labels on these disks as you create and process them.

### **NS2AL/CMD**

This program will run in Model I or III mode only, and will either convert one file at a time, or an entire disk, from NEWSRIPT format to ALLWRITE format. To start it, just type its name from DOS (not from BASIC):

**NS2AL (and press ENTER)**

The program will ask whether you want to retain NEWSRIPT control word conventions, or have them converted to standard ALLWRITE forms. NEWSRIPTS control words begin with periods, while ALLWRITE's begin with semi-colons (the semi-colon-period form no longer is needed). We recommend you switch to ALLWRITE's conventions, but you don't have to do so. It will not convert "!" Escape Sequences, but instead will add an ";ES !" at the beginning of the file so that your old ones will still work.

Next, you will be asked whether you want to convert a file at a time, or an entire disk. The default is "one at a time," and if you choose it, you'll be prompted for an input name and then an Output name. These 2 names must differ, since, in manual mode, the original files will not be replaced. If you choose "automatic" mode, you'll be asked for the name of the list containing the names of the files to be created. If the list was created by "DIRLIST/BAS", its name will be:

### **FILELIST/LST**

In automatic mode, each file will be read, converted, and replaced on the same disk, so you should use BACKUP disks when you run this way. Don't try any shortcuts: if you convert your only copy of a file, and something goes wrong, the file may be lost forever. Remember, there could be a power failure or disk error, not to mention the remote possibility of a software "bug."

Automatic mode is quicker and less effort than typing the names one at a time, but you can pick whichever method best meets your needs. The converted files may be a little larger than the originals, so if the input disk is almost full, there may not be enough room on it to store all the converted files. To avoid the problem, move some of the files to another disk, then remove (kill) it from the input disk, and try again.

File conversion is fairly fast. For example, a full, 40-track, double density Model III disk will be converted in about 10 minutes in automatic mode. You don't have to convert every NEWSRIPT file you've ever written, just the ones you are still using, or expect to use in the future. If you need to convert some others later on, you can do it then.

The conversion program handles most NEWSRIPT files, but if you've been using the Daisywheel Proportional Option, you should either copy some of the proportional tables from NS to the ALLWRITE disk, or change the ";TR" control words as described in "Proportional Printing on Daisywheels" in Chapter 5 of this book. The common tables are built right in now, so it will be easier and faster than before to access them.

This concludes the instructions for converting files.

### **DIFFERENCES BETWEEN NEWSRIPT AND ALLWRITE**

We wrote this section for people who already know how to use "NEWSRIPT," but it presents an overview of ALLWRITE that may be interesting to our new customers also.

ALLWRITE is written entirely in true, native, Z-80 machine language, and does not use BASIC. The Editor is similar to NEWSRIPT's in some ways, and very different in some other ways. Rather than list all the differences, we suggest you read the tutorial in the next chapter ... sooner or later, you'll break down and read it anyway, and it'll be easier for you (and us) if you do it sooner.

The Editor is very fast. It reads a 25,000-character file from floppy disk in 6 seconds, does a global change in less than four seconds. It has room for over 25,000 characters on a Model I or III (and over 34,000 or 99,000 on a Model 4). It can scroll horizontally if your text is wider than the screen, and joins short lines together automatically or when you press control "J" (CLEAR is still the control key, but if you're using a Model 4, you can use either CLEAR or F1). The "command line" isn't on the screen except when you press "BREAK", and then it appears at the bottom of the screen. The "Status" display includes a word count as well as several other useful values. There's no particular limit to the number of "lines" in a single file.(NEWSRIPT) had an upper limit of 400, after which an "APPEND" was needed); the only limitation is available memory, or your judgement that a new topic belongs in a separate file.

There's no "LIMA": its functions now are done by simple control keys, and we have no doubt that almost all of you will prefer the new approach within moments of seeing how it works. The new screen format uses the entire screen to display text, so a Model I or III shows 1,024 characters at a time, as compared with 960 in NEWSRIPT; a Model 4 shows 1,920 characters at a time. Text wider than the screen is easily displayed by instantaneous horizontal scrolling: text width simply doesn't matter anymore.

The Editor supports 22 "soft keys". "Soft keys" save typing time because they can be defined to stand for anything that can be typed in from the keyboard, including commands, escape sequences, control keys, words, phrases and even other soft keys. We've predefined some of them for your convenience, but you can (and will) change them around to suit your needs. Each soft key can store 22 keystrokes, and soft keys can be chained together if you need long stock phrases. Each group of definitions can be stored on disk, so you can switch from 1 set of 22 to another, resulting in an unlimited number of available keys.

The Editor can follow chained files, going forwards and backwards: the command is "LINK". When linking, the current file is saved (if it contains changes), and the next (or previous) file is read in. This gives the effect of being able to edit a file of unlimited size, and you control when disk reading and writing will occur. By using soft keys, a hard drive, and fast file reading, linking lets us search through this entire manual (75,000 words) in less than 10 minutes.

When the next file appears, all the edit settings you had been using will still be in effect, including line lengths, search and replace definitions, soft keys, screen grids, on-screen tab settings (thought you'd like to hear that), and anything else you may have set. We'll talk more about this in the tutorial. Once set, on-screen tabs can be stored on disk and loaded as you need them; and the print-time tabs are far more powerful than before.

The Text Formatter (the equivalent of "SCRIPT" in NEWSRIPT), processes text several times faster than SCRIPT did. It can store over two pages of text in its internal print buffer (and more than that on the Model 4), and uses most of the same control words and escape sequences that were in NEWSRIPT. This means your existing files still can be used (after conversion), and that you will have almost nothing new to learn until you want to use new features.

### Using NEWSRIPT Conventions

Other than improved speed and handling of short screen lines, the most obvious difference between using NEWSRIPT and ALLWRITE is that, by default, control words now begin with semicolons ";" instead of with periods; and escape sequences (now called "Emphasis Marks") begin with at signs "@" instead of exclamation marks. Also, multiple control words on a line are separated, just by the semicolons, instead of by semi-colon/period. These changes were made to save keystrokes and finger movement: the semi-colon is on the "home row" of keys for touch-typing, which makes it quicker, easier, and more accurate to use than the period was. However, if you want to keep using the period and the exclamation mark, just define them in the "xxx/DEF" file for your printer (it is created by the installation procedure), and you'll be able to type just the way you did before:

**;ES;CW**

"CW" stands for "Control Word", and lets you redefine the control word symbol. "ES" stands for "Escape Sequence", just as it did in NEWSRIPT.

Short lines on the screen will not stay short with ALLWRITE unless they end with an "ENTER" symbol: if there's no "ENTER", the lines will be joined together automatically to fill out the screen. Control word lines must be on screen lines of their own, and be preceded and followed by "ENTER" symbols. This was true in NEWSRIPT also, but since the Editor there didn't join lines unless you told it to do so, you never needed to be aware of this requirement. If you want ALLWRITE to work the same way as NEWSRIPT did in this respect, just use the command "JOIN OFF" when you start. If you don't like the shape of the "ENTER" symbol on the screen, you can change it during installation, or set it to "128" to make it invisible. We advise against that, but if you make those two changes and re-define ";CW", ALLWRITE will work very much like NEWSRIPT, only faster and with almost no restrictions.

List of Changed Control Words

The control words that have changed are listed below:

<u>OLD</u>	<u>NEW</u>	<u>COMMENT</u>
BF	PI	begin font now "PITCH"; 737 -> 0 (see "PITCH" in chapter 5 for details)
LS	LG	Logo Space
SS	LS	Line Spacing: 1, 1.5, 2, 3
DS	LS	*
TS	LS	*
SH	SK	can skip .5 lines (1/2 line), -4 lines, etc
SP	SK	use "SKIP" with the "A" (Always) option
HS		heading space, now automatic
FS		footing space, now automatic
HM	TM	heading margin now 2nd value of TM
FM	BM	footing margin now 2nd value of BM
PP	PP	paragraph simplified: indent, # blank lines
SD	SD	2nd value of Y or N, not 1 or 0
NU	NU	1st value swapped with Left Margin
TR	TR	"disk tables" usage completely changed (see "TRANSLATE" and "PITCH" in chap. 5 for details: many tables are built in now)

Other control words have been enhanced, new control words have been added, and the default escape sequence character is now the at-sign "@" instead of the exclamation mark "!", because it's easier to reach and isn't a shifted character. However, if you want to use the exclamation mark, "ES !" will take care of it for you, just as it did in NEWSRIPT. By the way, we now call those "Emphasis marks" instead of "Escape sequences", because that new term makes more sense and sounds less like computers. We've reversed the notation used for sub and super-scripts ("!+ used to mean "subscript", but now the "+" means "super-script", while the "-" means "sub-script"). Also, the Boldface escape sequence has more capabilities than before:

OLD: !\* (turns on boldface)  
!: (turns off boldface)

NEW: @ \*n 'n' is a number from 0 to 9  
0 = no boldface  
1 = one overstrike or hardware boldface  
2 = 2 overstrikes or hardware boldface  
3 = 3 overstrikes, (etc.)  
8 = eight overstrikes  
9 = shadow print (depends on printer)

Example: This should be in @\*2boldface.@\*0

Instead of having a dozen or so "Run-time options", almost all control words can be specified as options. When you type them in, they must look like a line of control words (each one must begin with a semi-colon), and be treated as such. All other "SCRIPT" features are still available in ALLWRITE, including an expanded Form Letters capability, Mini-Edit, and a greatly enhanced "output to video" capability for previewing a document and catching errors.

**MAILING LABELS OPTION**

NEWSRIPT's "Mailing Labels Option" has been replaced by ALLWRITE's built in ability to print side-by-side mailing labels. This is a standard part of Form Letters. Your NEWSRIPT Name and Address files will work as-is. "Super-lists" aren't needed (or supported) any more, since you can edit 400-500 labels in a single file. ALLWRITE isn't limited to any particular number of lines in a file, only to the size of memory.

**DAISY WHEEL PROPORTIONAL OPTION**

Proportional printing on certain daisywheel printers was an option in NEWSRIPT, but is standard in ALLWRITE. Most of the popular tables for wheels and thimbles have been built right into the printer tables, and the remaining tables are still on the distribution disk if you need them. If you made any tables of your own, they still will work with ALLWRITE. The "BUILDTAB" program is not included with ALLWRITE but you can use your old copy if you want to.

**PRIMARY OPTIONS MENU AND EXIT MENUS**

ALLWRITE doesn't use, or need, a "Primary Options Menu," because the Editor provides all the needed functions directly. Soft keys have been defined to let you run the Text Formatter (the equivalent of "SCRIPT"), Electric Webster, and DOTWRITER. Directories can be displayed when you are prompted for a file name, or by the "DIR" command within the Editor. Directories can be "masked" for selective displays, but no longer are alphabetized or numbered because of the excessive memory sorting would require. If you tell the Editor to run the Text Formatter without any options, (the old #1 choice with NEWSRIPT's Editor). You won't have to answer any prompts when printing finishes: the Editor will be run automatically, and will read in and display the file you've been printing ("untouched by human hands").

**USING ALLWRITE WITH ELECTRIC WEBSTER**

The Editor's Soft Key "Capital four" (Control dollar sign) runs the Electric Webster spelling checker, if you own it; otherwise, you'll just get an error message. (Electric Webster may be ordered directly from PROSOFT.) The versions of Electric Webster that worked with NEWSRIPT are not integrated with ALLWRITE (it didn't exist back then) and do not recognize the use of a semi-colon as the control word indicator. An update to EW is needed for maximum compatibility and convenience: you can order it directly from us, or from Cornucopia Software. If you're running in native Model 4 mode, you will need the Model 4 version of Electric Webster. When ordering the update, you must send in your original EW diskette.

**USING ALLWRITE WITH DOTWRITER**

If you have DOTWRITER 4.0 you can run it directly from ALLWRITE's Editor by pressing soft key "capital 3" (control pound sign). When DOTWRITER ends, you can return to ALLWRITE by pressing "3" on DOTWRITER's exit menu. The transition is smooth, fast, and convenient.

ALLWRITE still can be used to edit files for all versions of DOTWRITER, but cannot run versions earlier than 4.0 directly: you must exit from ALLWRITE and start basic yourself. If you have a version earlier than 4.0, we recommend you order an update to 4.0 from PROSOFT: 4.0 is much faster and more convenient to use than 3.0. We offer versions of DOTWRITER for native Model 4 operation as well as for Models I and III.

Regardless of the printer you're using, the appropriate DOTWRITER "/CMD" file must be renamed to "DOT/CMD" so that ALLWRITE and TCONINX can find it.

### **CHAPTER 3**

#### **LEARNING TO USE ALLWRITE**

The easy way to master ALLWRITE is by learning just a little bit at a time. This chapter will explain enough about ALLWRITE to let you write, print, and revise a letter. It will cover only a few features, but they will be the ones you will use most of the time. Afterwards, you can learn other subjects as you need them. To help you in this, most of the book is organized by "topic". For example, "setting margins" is a topic, and "deleting text" is another topic.

Many computer books present their features in alphabetic order. That's fine if you already know the names of the commands, but it isn't very helpful to a beginner. So, we've organized this book by topic, beginning with the most commonly used features. We've put a lot of synonyms for these commands into the index, so you can look things up without necessarily knowing the actual commands. The index is alphabetical, the Table of Contents is detailed, the summaries in chapter 8 contain cross-references, the Cue Card is right in front of you, and you can always press the "HELP" key to get needed information in a hurry. In other words, there are several ways in which you can find what you need.

ALLWRITE's capabilities are divided into two main parts; a "Text Editor" and a "Text Formatter." The Editor is used to type in text and make corrections to it. It is a "program" (a set of instructions telling the computer what to do) and will be the thing you mostly use when working with ALLWRITE. You can tell the Editor how to display or change the text in a file by using "Control Keys" and "Commands". A control key performs a simple function, such as deleting a letter. A command is something that is similar to the DOS commands. Most commands require parameters (words or numbers) such as file names, "ON", or "3".

The Formatter is used to print your documents or preview them on the screen to let you see how they will be printed. You tell the Formatter what to do by typing "control words" and "emphasis marks" into your text while using the Editor. A control word begins with a semi-colon, followed by two letters, such as "CE" for "Center". An emphasis mark begins with an at-sign "@" and is used for special effects such as underlining or boldface.

The trick to learning ALLWRITE is to realize that all its features are based on English-language "mnemonics." That means that it is controlled through a combination of: English words (such as "SEARCH"); one-key operations that use the first letters of English words (such as "D" for "delete"); and two letter mnemonics (such as "CE" for "center"). If you keep this in mind, you'll find that you already know ALLWRITE's commands, and will just have to find out which common words we've chosen. We have tried to pick obvious words and mnemonics that will be easy to remember, but whenever you get stuck, you can use the built-in "HELP" feature to get a quick explanation, or look up the topic by referring to the index in this book.

The list below summarizes the topics we will cover in this chapter. It may be all you will need to read if you're experienced in using word processors, but if you're brand new to this sort of thing, don't let the size of the list frighten you off: most of it is really pretty obvious.

#### **TOPICS IN THIS CHAPTER**

Starting ALLWRITE to edit a file: text entry the ENTER arrows shifted arrows scrolling: keys repeat simple corrections Control key control key functions	"ALK" or "AL", press ENTER type its name, press ENTER just type, automatic word wraparound key appears on screen as a backward "L" move cursor in 4 directions move cursor to 4 edges of screen arrows, control keys, or as needed if held down type right over the text CLEAR (also F1 on Model 4) shown as <>, with key in middle
--	---

**Text Deletion**

<D>	delete character under cursor
<W>	delete rest of word
<L> or <SPACE BAR>	delete rest of line
<S>	delete rest of sentence
<P>	delete rest of paragraph

**Text Insertion**

<I>	turns insert on or off
<O>	one keystroke insertion

**Editing Commands**

STATUS	shows size of text, etc
WHOOPS	discards recent changes
SAVE	writes text to disk file
HELP	displays the HELP menu
DIR	displays a disk directory
QUIT	exit to DOS without SAVE

**Text Formatting**

control words & emphasis marks	
;	CE
;	SK
;	PP
;	PA
;	IN 5
;	FO OFF
;	FO ON
;	PI 0,10,12,16
	center next line of text
	skips a line on paper (blank line)
	starts a new paragraph (soft key 1)
	starts a new page (soft key 2)
	indent left margin 5 spaces
	print text as-is, ragged right margin
	print text with fully justified lines
	pitch; 0=proportional

**Emphasis Marks** within text (Escape sequences)

@\$	start underline (soft key 4)
@%	stop underlining (soft key 5)
@*2	start boldface; any number, 1-9
@*0	stop boldface (soft key 7)
@(	starts double width (dot matrix only)
@)	stops double width (dot matrix only)

**Formatted Printing**

@/ <!>	Control Shift 1 or 2
<">	formatted printing, no options formatted printing, ask for options

**STARTING ALLWRITE**

To use ALLWRITE, a "DOS" disk must be in drive 0, the ALLWRITE "working disk" must be in a drive (the working disk usually will be combined with the DOS disk to make more efficient use of disk drives), and a formatted "data disk" should be in drive 1. If you have a one-drive computer, then the "data disk" will be a minimal DOS disk with as much empty space as possible, and will be put into drive 0 after ALLWRITE starts.

With the necessary disks in place, press the RESET button on the computer. On the Model I, this is behind the keyboard on the left side; on the Models III and 4, it is the orange button on the right; and on the 4P, its to the left of the screen. When the DOS indicates its ready, type 1 of these 2 commands and press the ENTER key:

**ALK**                    (if you're using our keyboard driver)  
**AL**                    (if you're not using our keyboard driver)

If you use "ALK", it will start "AL" for you. If the DOS command "AUTO ALK" or "AUTO AL" is in effect when you boot the DOS disk, ALLWRITE will start by itself.

After a moment or two, the name "ALLWRITE" will appear in big letters on the screen; shortly thereafter, this message will appear at the bottom of the screen:

**ENTER NAME OF DOCUMENT TO BE EDITED ==>**

This kind of message is called a "prompt" because it explains the kind of answer that it expects you to give, and then waits for a reply. The "name" of the document can be anything you like, as long as it conforms to the requirement of your DOS. In general, a file name on a TRS-80 consists of a word, possibly followed by a slash "/" and a three character extension. If you aren't sure how files are named, please see the explanation in your DOS manual.

Reply to this question by typing a valid file name and then pressing the ENTER key. If the file already exists, it will be brought into memory and displayed on the screen. If you want to create a new file (a letter or document), then choose a new name; for tutorial purposes, we'll pick "TEST" and then press ENTER.

If the file whose name you choose is not found, (If its a new file, it certainly won't be found), this message will appear:

**THIS WOULD BE A NEW FILE, PRESS ENTER TO CREATE IT,  
 MINUS "-" TO CANCEL, "?" FOR A FILE DIRECTORY, OR  
 TYPE ANOTHER FILE I.D. IN ALL CASES PRESS "ENTER" LAST**

For now, just press the ENTER key. Please note that you've assigned a name to what you're about to write, and ALLWRITE will remember that name for you.

When you press ENTER, the screen will go entirely blank, and a blinking "cursor" (the small square that shows you where you are on the screen) will appear in the upper left-hand corner. If you had selected a file that already existed, the beginning of it would have appeared on the screen, and the cursor still would have been placed in the upper left-hand corner.

**HOW TO USE THE EDITOR**

When we write by hand, each sheet of paper usually is 8-1/2 inches wide and 11 inches long. When we reach the end of one page, we just continue on the top of another page. When we write with a computerized word processor, the "page" can be a lot longer than a piece of paper, and a lot wider, too. However, because the computer's screen is of limited size, we can see only a little of this huge "page" at a time. It may help to think of the screen as a "window" through which our words can be seen. The window can be moved up and down, left and right, as necessary, to see other parts of the document. ALLWRITE offers several ways of moving that window around.

**UPPER AND LOWER CASE**

Let's begin by just typing the first two lines of the paragraph you are reading right now. If everything appears in UPPER CASE, you can switch to lower case by pressing SHIFT ZERO (that's the standard way it's done on a TRS-80). On a Model 4, you can press the "CAPS" key instead. If you're using a Model I and it doesn't have lower case, you will have to type in upper case, or your text will appear as meaningless characters.

**WORD WRAP AT RIGHT EDGE OF SCREEN**

As you type, you'll see that the cursor keeps moving to the right. When it reaches the right edge of the screen, it will move to the start of the next line on the screen, taking with it the word you were just typing. This is called "word wrap", and it means you won't have to press ENTER to get to the next line, and that a word won't ever be split across two lines.

If you set the length so that it's wider than the actual screen, and the cursor reaches the right edge, the text will start to shift to the left as you continue typing. Word wrap won't occur until the cursor reaches the length that you've set. If you set line length so that it's narrower than the screen, word wrap will occur when that short length is reached. The way to change screen size will be covered later.

If you follow the convention of leaving 2 spaces between sentences, the Editor will try to keep both of them. To do this, it will check the next word you type on the screen. And if it cannot fit the next word on the line, it will move both the first word of the new sentence, and the last word of the previous sentence, to the next line -- a fancy kind of "word wrap." When this happens, a screen line may not be fully filled out, but everything will still print properly, including your double blank. If you don't leave two spaces between sentences, the Editor will allow the last word of a sentence to be the last word on a screen line.

**TYPEAHEAD**

"Typeahead" is a feature that lets you keep typing without losing keystrokes while the computer is still finishing the last thing you told it to do. It's a particularly important feature for a Word Processor, because you may not be looking at the screen as you type, and certain things can happen that will take more time than the brief interval between your keystrokes. That's why we supplied you with "ALK/CMD": if your DOS doesn't have its own Typeahead capability (and most of them don't), you should use "ALK" to make sure you won't lose keystrokes. As a secondary benefit, our "ALK" lets keys "repeat" much faster than they would with most DOS's.

There are three things to be careful about when using Typeahead: 1) ALK can store up to 128 keystrokes if necessary. That's two full screen lines on a Model I or III, and should be enough even for a speed demon. 2) A more important consideration is that Typeahead can be disabled temporarily by your DOS: when floppy disk drives are actually running, you can lose keystrokes. There's nothing we can do about this, since the DOS doesn't let ALK function when certain disk operations are in progress. So, when the disks start to turn, just sit back and wait for them to finish. 3) "DO" files may not work with "ALK", since they both need to use the same facilities of the computer. If you're in DOS with ALK active, and start a DO file, and it "hangs up", you may be able to get out of the problem by typing "ALKOFF" if that program is on-line at the time.

The problem with losing keystrokes when disk drives are running rarely happens with LDOS, TRSDOS 6.x, or when you use a hard drive. Of course, if you're using LDOS or TRSDOS 6.x, you will also be using the keyboard support included with your DOS, so you can just keep typing while text is being saved to disk or another program is being brought into memory for execution. Note: in TRSDOS 6.2, the "SMOOTH" parameter of the "SYSTEM" command can cause loss of keystrokes when the disks are running.

**THE "ENTER" KEY**

When you reach the end of a line on a typewriter, you must press the "ENTER" (or "RETURN") key to start another line. That isn't true with ALLWRITE, since it moves to the next line automatically. This means that the "ENTER" key is used for special things: if pressed in the middle of a line of text, it just moves the cursor to the start of the next line,

just like a typewriter. If pressed at the end of a line, however, it does something extra: it marks the end of something, usually a paragraph, and puts a special symbol on the screen to show this has happened. The symbol looks like the "RETURN" key on an IBM typewriter, a backwards "L". To see what we mean, if the cursor is now past the last letter on the current screen line, just press ENTER. We will refer to that special symbol as the "ENTER" symbol. For those of you with some word processing background, it marks the end of a block. If you don't know what that means, we will be covering it later.

If you're at the end of a line, and want to move to the start of the next line, but don't want ALLWRITE to add an "ENTER" symbol, press "SHIFT RIGHT ARROW" instead of "ENTER".

So far, what you've done is no different from using a typewriter, except for not having to press ENTER. Now with a couple of lines on the screen, we can learn about some of the "editing" features that make a word processor so nice to use.

### **USING THE ARROWS TO MOVE THE CURSOR**

First, the four arrows will move the cursor around without erasing any text. Try pressing each of them to see what happens. Next, if you hold down any key, including the arrows, it will start to "repeat". This makes it very convenient to move the cursor along a line by holding down the left or right arrow.

If you hold down the UP arrow, the cursor eventually will reach the top of the screen. If you keep holding it down, the text you've typed will move DOWN one line, and then all movement will stop. You now have a blank line at the top of the screen. If you don't type anything on it, ALLWRITE will throw it away later on, but if you do type anything there, it will become the new first line of your document. It's an easy (and the only) way to add text above the first line of a file.

If you hold down the DOWN arrow until the cursor reaches the bottom of the screen, and keep holding it down, the text will move UP, and start to disappear off the top of the screen. This will continue until you let go of the arrow, or until the last line of text reaches the top of the screen. When it does, no further movement will occur.

### **SCROLLING**

The movement of text across the screen is called "scrolling." When you have more than a screen full of text, you can move it up and down just by holding down the arrows. Other ways to move around in the text will be covered later. If you set the line length so that it's wider than the screen, the left and right arrows will let you "scroll" from side to side. When the left/right arrows cause the cursor to reach the sides of the screen, the cursor will just move to the next or preceding line.

### **OVERSTRIKING TEXT**

The editor offers you many ways of making changes and corrections. One of these is similar to what you would do with a typewriter or a pencil that has an eraser: you can use the arrows to move the cursor until it is on a letter you want to change, and then you can press the correct letter. When you do this, the new letter will replace the old one, and the correction will have been made cleanly. That's a lot nicer than erasing, or even using lift-off tape.

We will call this kind of correction "overlaid" text, and the square shape of the cursor (unless you changed it) was chosen to remind you of a square, lower case "OH": it's a reminder that you are in "overlay mode," which is the normal way to use the editor.

**QUICK CURSOR MOVEMENTS**

So far, we've moved the cursor either by typing or by pressing the arrows. In both cases, the cursor moved only one position at a time. Sometimes, you will want to move the cursor farther than that, and quicker. Holding down the SHIFT key while pressing an arrow is one of the ways to do this. The list below shows what will happen:

SHIFT LEFT	cursor to start of line if in middle or to start of preceding line if at start
SHIFT RIGHT	cursor to end of line if in middle or to start of next line if at end
SHIFT UP	cursor to top of screen if in middle or scroll one screen back if at top
SHIFT DOWN	cursor to bottom of screen if in middle or scroll one screen forward if at bottom

Try each of these now, if "SHIFT DOWN ARROW" doesn't work, you aren't using "ALK". In that case, use "SHIFT DOWN ARROW Z" instead. (Sorry, but that's how some DOS's as well as the hardware of the Models I and III work, and it's one of the things "ALK" fixes.)

**THE CONTROL KEY -- CLEAR**

You probably know that a word processor should be able to do lots of other things besides accept text and simple cursor movements, and that some computers have extra keys for "inserting" and "deleting" text. The TRS-80 doesn't have those keys, and even if it did, there are dozens of other conveniences that should be in a word processor. To provide a dedicated, labeled key for each of them would just about double the size of the keyboard.

Fortunately, there's a better way to solve the problem. Just as the "SHIFT" key doubles the number of characters you can type, a "CTRL" key can give you what amounts to a third complete set of keys. Needless to say, ALLWRITE has a "control" key, the one marked "clear". On the Model 4, you can use the "F1" instead; and on the LNW-80 you can use "CONTROL" instead. The "CTRL" key on the model 4 and the MAX-80 cannot be used, because the DOS uses that key to modify all the other keys on the keyboard. If you press "CTRL" and the letter "D", for example, DOS will not pass the "D" to ALLWRITE; it will pass a binary "4", which isn't even the "4" key! "F1" on the MAX-80 will set control mode, but holding it down while pressing another key will cause that other key to appear on the screen, so it should not be used for this purpose.

Regardless of which key you use, we will call it the "control key" from now on. Please remember this, because we won't mention "CLEAR" or "F1" again, only "CONTROL". When you press it, the cursor will change from its normal "OH" shape to a large, graphic "C" (for "Control", of course). Once pressed, you don't have to keep holding it down: it "latches" for one keystroke. You can hold it down while pressing another key, if you wish, but that's up to you.

Note: With LDOS and TRSDOS 6.0, certain keys will not repeat if held down along with CLEAR. Just use "F1" instead.

When the Control cursor is on the screen, the editor is in "control mode," another term we will use from time to time. If you just want to get out of control mode, press Control again; the cursor will return to its previous shape. Otherwise, you can press another key to tell ALLWRITE to perform a special function.

**DELETING TEXT-- <D> <W> <L> <S> <P>**

The 3 most common kinds of corrections you'll be making in a document are overlaying text, deletions, and insertions. We've already explained that just moving the cursor to the old text and typing right over it does overlaying. Deletion and insertion are done with control keys.

To delete a single character, move the cursor so it covers that character, press CONTROL, then press the letter "D" (for "Delete"). The character will disappear, and everything to its right on the same line will move one position to the left. If you hold down both CONTROL and "D", more characters on the same line will be deleted: they will seem to be falling into the cursor. When there's nothing more on the line, deletion stops, so you won't have to worry about overshooting.

Saying "Press CONTROL," then press "D" will get a bit tiresome after a while, so we will abbreviate it from now on by putting the "D" in brackets, like this: <D>. In the remainder of this book, whenever a letter or symbol is surrounded by those pointed brackets, it will mean that the Control key, and then the indicated key, should be pressed.

<D> should be easy to remember, because "D" is the first letter of "Delete." You will find that most control keys have been matched to the first letter of the functions they perform. For example, there are several other kinds of deletion available: Word delete, Line delete, Sentence delete, and Paragraph delete. The control keys for these are <W>, <L>, <s>, and <P>. See? You just learned 4 more functions -- but you already knew them; all we had to do was confirm to you that ALLWRITE had those capabilities.

When any of these deletions is done, the character covered by the cursor, and all characters to the right of the cursor, up to the limit you've chosen, will be deleted. So, if you cover the "h" of "the" with the cursor and press <W>, "he" will be deleted, but the "t" will remain. If you press <L>, the rest of the screen line will be deleted, but nothing to the left, and nothing on the following lines, will be affected. Pressing Control and then the Space bar does the same thing as <L>: it deletes the rest of the line. This use of the space bar was provided because it's easier to reach than <L>, and to maintain compatibility with NEWSCRIPT.

<D>, <W>, and <L> only affect the contents of one line, so if you press one of them by mistake, not too much damage can be done. In fact, there's a way to recover from this kind of mistake, and we'll get to it soon. But, if you press <S> or <P>, a lot of text might be deleted, and that isn't safe. So, whenever you try to delete more than one line of text, the editor will display a message that tells you how much will be deleted. If it sounds about right, you can confirm the deletion by pressing the letter "Y" (lower or upper case, don't press ENTER afterwards). If you press anything else, the deletion will be cancelled. This is one of ALLWRITE's safety features.

<W> leaves a blank between words, whether you delete an entire word or just the end of a word. However, any punctuation at the end of a word will be deleted along with the word itself.

<S> deletes up to, but not including, the next period following the cursor. This can lead to unexpected results if the sentence contains something like "Mrs. Smith" or "\$9.95": deletion will stop before the period in either of these cases.

If you use <S> from the middle of a sentence, the results will be just what you want, since the ending period will be retained. However, if you want to delete an entire sentence, you'll have to delete the period and extra blanks with <D>. <S> doesn't delete them for you because it can't be sure whether you're deleting part of a sentence or the entire sentence. If you want to delete the entire second (or third, etc.) sentence of a paragraph, an efficient approach is to position the cursor on the period of the preceding sentence, then press <S>.

<P> deletes up to, but not including, the next "ENTER" symbol on the screen. The "ENTER" is kept because it's the way you indicate where one thing ends and another begins.

Another way to delete the rest of a line is with <SPACE> (Control-space bar). It does exactly the same thing as <!.> and we provided it for compatibility with NEWSCRIPT.

### INSERTING TEXT -- <I>

Insertion is the opposite of deletion. If you've left out a letter, a word, or more, you will want an easy way of adding what was omitted. To do so, use the arrows to move the cursor to the position where you want to insert text; there will already be something there, of course. Then press <I> (control "I" ... I for "insert"). The cursor will change shape so that it looks like a capital "I", tall and thin. Now, you can type the letter(s) that were missing. As you do so, the text under and to the right of the cursor will slide to the right, making room for the letter(s) you've just typed. When you're finished inserting, press <I> again to return to the overlay mode.

Inserting A Single Character -- <O>

Unlike <D>, <I> stays in effect until you press <I> again to turn it off, or until certain other things happen to allow ALLWRITE to turn "insert mode" off for you. That's very nice for typing in missing words and sentences, but very often, you will only need to insert a single letter. It would be a bit inefficient to have to press a total of five keys just to insert one letter (control, I, the letter, control I). So ALLWRITE offers you a shortcut: <O> (that's the letter, "OH") stands for "One key insert", and puts ALLWRITE into Insert mode only for one keystroke. As soon as you press one letter (or any character), it will be inserted at the cursor, and ALLWRITE will go back into overlay mode.

MOVING TO THE TOP OR END OF THE TEXT FILE -- <T> <E>

When a document becomes large, using the arrows to scroll through it can become tedious, even at the high speed used by ALLWRITE. There are some better ways to do this, and two of them are control keys that move the cursor directly to the top or end of the file. By now, you should be able to guess what those keys are <T> and <E>. (We called it "End" instead of "Bottom" because <B> stands for "BLOCK", a feature we haven't covered yet.)

<E> positions the text so that the last eight lines are shown, and the cursor is placed on the first blank line after the text. It's done this way to let you see some of the last words you wrote, while leaving room on the screen to add more text without immediately having to scroll.

To summarize the control keys we've learned so far:

DELETION

<D> delete character  
 <W> word delete  
 <L> line delete (SPACE bar  
     can be used instead)  
 <S> sentence delete  
 <P> paragraph delete

INSERTION

<I> insert mode on or off  
     (goes ON if it was off,  
     or OFF if it was on)  
 <O> one-key insert

SCROLLING

<T> to top of text  
 <E> to end of text

Although they won't be covered for awhile, we should mention that there's one more "delete" available, and one more "insert." Since we're only trying to teach you enough to get started with ALLWRITE, we will leave those features for later.

THE COMMAND KEY – BREAK

A "Command" is something you type in order to tell a computer what to do next. The Editor's Commands are something like the ones you use with your DOS: they are words, often followed by other words or numbers called "parameters", which is just a fancy way of saying that commands can be modified by what comes right after them. "DIR :1" is a DOS command followed by one parameter, and it happens to be an Editor command, also.

Your DOS treats everything you type as a command, but that wouldn't make sense with an editor, because you mostly will want to type text, not commands. Instead, when you want to give ALLWRITE's editor a command, you must press the "BREAK" key and then release it. Since you aren't using "BASIC", this won't stop or terminate the program; it only will signal the editor to expect a command from you.

When you press "BREAK", the bottom line of the screen will temporarily be replaced by this prompt, which we will call the "command line":

**CMD=>**

Try it now to see what happens. If you don't really want to give a command, just press "BREAK" again, or press the "ENTER" key. The prompt will disappear and whatever text had been on the bottom line will re-appear.

### The STATUS Command (Also Control 9)

Although the editor has dozens of commands, you will use only a few of them frequently. Let's start with one called "STATUS". Press "BREAK" so that the "CMD=>" prompt is on the screen, and notice that the cursor has been moved so that it immediately follows the prompt. Now, type "STATUS" and press the "ENTER" key. When you do so, a screen full of information about your text file will be displayed. It will look something like this:

ALLWRITE (tm)	FILE STATUS	FOR	AREA 1
CHARS USED: 1147	WORD COUNT: 187	PRINTER1: FX80	
CHARS LEFT: 24053	LINE COUNT: 31	# CHANGES: 47	
TOTAL SPACE: 25200	SCREEN WIDTH: 80	AUTOSAVE: 65535	
LAST FILE I.D.: (none)			
CURRENT FILE I.D. =====> TEST			
NEXT FILE I.D.: (none)			
SEARCH =>			
REPLACE =>			
PRESS <ENTER> TO CONTINUE....			

The size of your text is shown in characters, words, and screen lines. The amount of space still available in memory is shown, too. The word count excludes control word lines and is useful for writers and students; the line count refers to the number of "screen lines" the text would occupy if you had a big enough screen. It doesn't refer to the number of lines that will be printed by the Text Formatter, although it is usually a good approximation. The current screen width is shown, and probably will be "64" or "80".

Also shown is the printer you're using, the number of changes you've made to the file since it was read in or most recently saved, and the "AUTOSAVE" threshold (we explained AUTOSAVE during Installation). The name of the file you're editing is shown prominently in the center of the screen. The "LAST" and "NEXT" file names will be "(none)". Later in this book, we'll explain how text files can be "linked" together to let you write documents of unlimited length. When you do link files together, the names of the two immediately adjoining links will be shown in "LAST" and "NEXT". "AREA 1", shown in upper right hand corner, is meaningful only on a 128 Model 4, and will be covered later on.

When you've finished reading the STATUS screen, you can return to editing by pressing "ENTER". Your text will re-appear, and the "CMD=>" prompt will be gone. That's how all commands work: when they're done, they return you to editing, unless you used a command that ends editing. If you want to print the status screen, see below.

By the way, if you've placed the "soft key" strip above the keyboard, you probably noticed that the word "STATUS" appears just above the number "9". You may be wondering whether it would have done the same thing for you as the "STATUS" command. The answer, of course, is "Yes": if you press the CONTROL key and then "9", the status screen will be shown. However, the objective of this tutorial is to teach you how the editor works, so we'll save the shortcuts for later. Now that you know there's a "STATUS" command and a "STATUS" key, you'll probably just use <9>, but you will also know that, when you do so, the editor is giving itself a "STATUS" command.

When you press certain soft keys, the bottom line of the screen will flicker momentarily, because the soft key has issued a command to the Editor. This is normal.

**SCREEN PRINT**

In chapter 4, we'll show you how to print part or all of your text file "as-is." If you want to print the Status screen or a Directory display, then press the asterisk "\*" instead of "ENTER" when you see the "PRESS ENTER TO CONTINUE" message.

**WHOOPS**

No, we didn't just make a mistake. "WHOOPS" is an Editor command, and a very useful one. It's used to throw away recent changes that are still on the screen, but have not yet been recorded into memory. To see how that works, type a few words on the screen now, then press "BREAK", type "WHOOPS", and press "ENTER". See? Those words disappeared. Try again, this time using <D> to delete a couple of letters, or <I> to insert something in the middle of other text. Now, press "BREAK" and type "WHOOPS" again. Those changes are gone.

"WHOOPS" is the command to use when you've accidentally typed over something you meant to keep, or inserted or deleted something and then changed your mind. You should know that "WHOOPS" doesn't work for everything: once text has scrolled off the screen, it is recorded, and any changes in it no longer can be removed by "WHOOPS". Also when you issue most commands, or use certain control keys, the editor has to record the screen into memory, which puts the text beyond the reach of "WHOOPS". By the way, this can be abbreviated to just "W".

Of course, recording in memory isn't the same as recording onto disk, and until your text has been written to disk, you can always prevent the changes from being saved permanently. However, most of the time, you'll be more concerned with making sure that text does get saved to disk, since storing text on disk is the only way to preserve it once you stop the Editor or turn off the computer. That's what we well cover next.

**SAVING TEXT TO DISK -- SAVE**

We said the Editor uses natural, English words for its commands, so, as you might expect, the command used here is "SAVE" (not very original, but certainly easy to remember). Since it is a command, press "BREAK" first, then type "SAVE", and press "ENTER". When you do this, the screen will go blank for a moment, and then this message will appear:

**(AUTO)SAVING test**

When your file has some name other than "test", the name you're using will not be shown. The disk drives will start to run, and the file will be written to whichever drive your DOS selects. When this is complete, your text will re-appear on the screen and you can continue editing. It takes only a few seconds to save text, and we recommend that you use this command frequently, not just when you've finished six hours of steady writing.

"SAVE" has some other capabilities, and they will be covered in the "FILES" section of chapter 4. For safety, its minimum abbreviation is "SA" not "S".

**HELP (Also Control Zero)**

ALLWRITE has a built in "HELP" feature. To use it, "HELP/AL" must be on the working disk, or on some other disk that's in a disk drive now. If it is, you can display information about ALLWRITE by using the "HELP" command. Of course, you've probably noticed that the word "HELP" is on the soft-key strip, just above the zero. Let's take the easier approach now: press the CONTROL key, and then the number zero. When you do so, the disk drives will start to run, a list of topics will be displayed, and the cursor will be next to the first topic.

To get help on any particular topic, just use the arrows to move the cursor across and down to the subject you're interested in, and then press "ENTER". The disks will run again, and an explanation of the topic in question will appear on the screen. If it answers your question, just press "ENTER" after you've read it, and your text will re-appear on the screen. If the topic wasn't what you expected, press the question mark "?" and the HELP menu will re-appear: The cursor will be at the left hand side of the row containing the last topic. If the cursor isn't next to a topic when you press "ENTER", a "Disk Error" message will appear, but no harm will be done.

**NOTE:** The UP and RIGHT arrows "wrap around" the HELP menu screen, staying in the line or column. The LEFT and DOWN arrows do not wrap around, but stop at the margins.

To give you some practice with "HELP", and to give you an idea of how much is available in ALLWRITE, press <0> to get the HELP menu, then move the cursor to "COMMAND LIST", and press "ENTER". After looking that list over, press the question mark to return to the menu, move the cursor to "CNTRL KEY LIST", and press "ENTER". As you can see, there are a lot of capabilities, but they are easy to remember because they are "mnemonics" of natural words. Also, since the HELP key itself is within easy reach, you can always get a reminder of just how a rarely used feature works. The two lists you just saw are backed up by other HELP topics that explain each command and control key in more detail.

### **DISPLAYING A DISK DIRECTORY -- DIR**

When you're using DOS and want to find out what files are on a disk, the command to use is "DIR". That same command is available in ALLWRITE. In addition, when prompted by the Editor or the Text Formatter for a "Document I.D.", you can display a directory by typing a question mark "?" instead of a file name. If you follow "DIR" or "?" by a drive number, the files on that disk will be displayed; otherwise, the default drive used by your DOS will be displayed.

It works with most DOS's, but not with some versions of MULTIDOS; and it will not work with disks whose Directories are password protected. When used with most DOS'S, the number of sectors used by each file will also be displayed; if this value exceeds '99', a pair of asterisks "##" will be displayed instead. Each sector contains 256 characters.

If you have a hard drive or an 80-track diskette, not all the names will fit on a single screen. In this case, the editor will stop when the screen fills, and wait for you to press "ENTER" before continuing. If you don't want to see the rest, press "BREAK" instead. If you want to print what's on the screen, press the asterisk "\*" instead of "ENTER".

When the directory display is finished, the editor will wait for you to press any key before returning to the text screen.

ALLWRITE's "DIR" command can be selective in what it displays. If you only want to see the files with the extension "LET", you can give the command this way:

**CMD=> DIR /LET:1**

If you want to see all the components of ALLWRITE (they all start with "AL"), you could type:

**CMD=> dir al**

That assumes they are on the DOS's default drive, probably drive zero.

This generalized selection is sometimes called "masking", and sometimes "wildcard": we'll call it "masking." It's quick and efficient, but assumes you've been using some organized naming conventions for your files. If you have been picking names randomly, this would be a good time to start using prefixes or extensions that will group your files by related category.

ALLWRITE doesn't sort (alphabetize) the directory before displaying it because that would take between 500 and 5000 characters away from the "text buffer" (the space available for editing text). The "masked" approach is a lot more efficient, and since it displays fewer names, is often easier to read.

**EXITING FROM ALLWRITE - QUIT**

When you're finished using the Editor, you can exit from it in any of several ways. One of these is to have ALLWRITE print what you've written, and this will be covered in the next section. If you want ALLWRITE to terminate and return to DOS, you can use this command.

**CMD=> QUIT**

It can be abbreviated to "QU", but not "Q". If any changes have been made to the file, but not yet saved, ALLWRITE will give you a chance to save the updated file first:

**15 CHANGES NOT YET SAVED TO CURRENT FILE  
REPLY "S" TO SAVE, "Q" TO QUIT ANYWAY,  
OR "-" TO RESUME EDITING**

The number "15" is just an example. This message will appear whenever you try to exit from the editor with unsaved changes. If there are no changes, the exit will be done without the warning message. If you press the letter "S" (not followed by "ENTER"), the file will be saved, and then the exit will be completed. If you press any other key, the exit request will be cancelled and your text will be re-displayed.

Some DOS's do not accept commands in lower case. If this happens to you, just press SHIFT ZERO to switch to upper case after exiting from ALLWRITE.

**EDITING ANOTHER FILE -- NEW**

When you've finished working with one file and want to edit or examine another one immediately, just issue this command:

**CMD=> NEW**

It is similar to "QUIT", but re-starts the Editor instead of exiting to DOS. If the current file has any unsaved changes, you'll be given a chance to save them, then you'll be prompted for the name of the file to be edited. The default will be the one you just finished. When you begin work on the next file, all you're Edit settings (search, replace, soft keys, tabs, etc.) will still be in effect.

**FORMATTED PRINTING**

So far, we've been showing you how to create and modify ("Edit") text. Now we'll show you how to print it. When we write a letter, we usually observe certain conventions: your address and the date appear first, towards the upper right hand side of the paper. The name and address of the person to whom you are writing appears next on the left side. The salutation "Dear ...." comes next, with one or two blank lines before it and one or two blank lines after it. The first line of each paragraph is indented, and there may be a blank line between each paragraph. The signature area appears in the lower right hand part of the page.

You could write a letter on the screen this way with ALLWRITE, but it would be a lot of work, and would place severe limitations on ALLWRITE's ability to help you make that letter look good when it gets printed. It would also be hard to revise, especially if you decided to move a lot of things around.

ALLWRITE uses a document formatting method called "Control Words", and a method of indicating underlining or other special effects called "emphasis marks" (if you're a NEWSCRIPT user, these used to be called "escape sequences"). A "control" word is a two-letter mnemonic, preceded by a semi-colon to signal that its not just part of the text. Control words must appear on screen lines of their own, and those lines must be preceded and ended by "ENTER" symbols. An "emphasis mark" is a special two or three character sequence that starts with an "at sign" "@". Although the semi-colon and at-sign are used for special things, you can also use them normally, as you have seen us doing in this book. If you don't want to use those two symbols, you can tell ALLWRITE to use something else.

This method has some advantages and some drawbacks. Among the advantages are:

1. It's fairly easy to learn ("CE" means "center"; "pp" means new paragraph);
2. It's very flexible and extensive: you can produce just about any combination of formatting results you need with ALLWRITE, and not to have to work very hard in the process;
3. The formatting markers are clearly visible: the control words appear on the screen along with the normal text, so you will always know what setting is in effect;
4. It's easy to change formats at any time, since the controls can be edited exactly the way any other text is edited. This means you don't have to learn two separate editing systems;
5. It is well adapted to the limitations of the TRS-80 screen.

The major drawback to the method is that the screen is not an exact image of what will be printed. However, the TRS-80 doesn't have the screen hardware that is needed to show "what you see is what you get", so some drawbacks must exist regardless of how formatting is controlled. It's true that, if you just wrote in 10 pitch (PICA), never used proportional pitch, underlining, boldface, sub or super-scripts, italics, or "boilerplate", then the screen's limitations wouldn't matter; but if you hadn't planned to use any of those things, you probably wouldn't have bought ALLWRITE in the first place.

Some "graphics" boards are available for the TRS-80, and some "work-alike" computers, such as the LNW-80 have graphics capabilities. The reason we don't use such hardware to produce screen images is one of economics: those systems account for a very small portion of the market, so it would have been prohibitively expensive for us to develop a special version of ALLWRITE for so few potential customers. Our apologies to those of you with those machines: it may give you some consolation to know that we own two LNW's and a MAX-80 ourselves, and that the entire Text Formatter was written on an LNW-80/2. We wanted the capability for our own use, but simply could not justify the time and memory it would have taken.

### On-Screen Formatting -- Preview Mode

Since we couldn't change the TRS-80 hardware, we gave ALLWRITE some marvelous ways of getting around the limitations of the screen. When you are doing simple things, you can set the screen's line length to match the paper width (the command is "LEN", followed by a number, such as "60"). More importantly, the Text Formatter, which we are about to discuss, has a superb "preview" mode that can display formatted output on the screen, without using the printer. It shows indentations, page breaks, titles, centering, and underlining; and marks text that will be printed with "emphasis marks" such as boldface, double width, and italics.

### RULES FOR USING CONTROL WORDS

There are some rules to remember about using control words:

1. They go on lines of their own: "control word lines"
2. They start with a semicolon ";"
3. The semi-colon must be at the left edge of the screen
4. The preceding line must end with an "ENTER" symbol. (This does not apply if the first line of the file is a control word line)
5. Control word lines must end with an "ENTER" symbol

In many cases, several control words may be typed on the same line, separated by their semi-colons. However, while learning ALLWRITE, you may want to use only one control word per line.

**FUNDAMENTAL CONTROL WORDS**

Seven control words and three pairs of emphasis marks provide the most common formatting capabilities you are likely to use. To make it even easier, seven of these 13 functions are on the Soft Key strip, clearly labeled for you. ALLWRITE has several hundred formatting capabilities, so when you need something special or fancy, it'll probably be there; but for writing a letter, you probably won't even use all the fundamental ones.

These are the seven fundamental control words:

:CE	center next line of text
	;CE 3 would center the next 3 lines
:SK	skip a line (leave a blank line)
	;SK 2 would leave two blank lines
:PP	start a new paragraph
	it leaves a blank line and indents 5 spaces
:PA	start a new page
:IN5	indent 5/10 of an inch
	;IN 0 restores to the left margin
:FO OFF	turns off part of text formatting, so what follows is printed "as-is"
	;FO ON turns formatting back on
:PI10	causes printing in 10 pitch (pica) Other pitches are: 12, 16, & 0 (proportional) They work only if your printer has that pitch. This control word was ".BF" in NEWSCRIPT, and pitch "0" was "737" in NEWSCRIPT. See Chapter 5.

These are the fundamental Emphasis Marks (Escape Sequences):

@ \$	turns on underlining, if your printer can do it
@%	turns off underlining
@ * 2	turns on boldface level 2 Other levels 1-9
@ * 0	turns off boldface
@ (	9=shadow print on some daisywheels
@ )	turns on double width Dot matrix only
;in0;hi0	turns double-width off

If you look at the Cue Card that fits just above the keyboard, you can see that Soft Keys 1-7 already have been defined to stand for the most useful of these control words and emphasis marks, and you'll find it very easy to control formatting by using them.

There's one error that's easy to make with this formatting approach: If you forget to turn something off, you'll get a lot of underlining, boldface, etc. that you didn't really want. That can be detected easily if you use the "preview" feature before printing the text; we will want to do that soon.

**SAMPLE LETTER**

A good way to illustrate the use of ALLWRITE's formatting language is by showing you a letter, first it would be written with ALLWRITE, and then it would be printed. The letter appears in these two forms on the next two pages, and is also on the distribution disk, under the name "SAMPLE/LET".

Please look the letter over, and see how the format controls affected the printing. Notice that the printed lines have smooth right margins, and that the lines don't correspond exactly to what was typed. ALLWRITE took care of most of the formatting so regardless of how it was typed in, it'll be printed nicely. For practice, and to use the text part of this tutorial, we suggest you type the "unformatted" letter in now... it's pretty short. If you don't want to type it, you can have the Editor use "SAMPLE/LET" from our disk.

This example begins with two control words that are not shown on the fundamental list; ";CM" stands for "COMMENT", and anything after it, up to the next "ENTER", will be ignored. ";LL" stands for "Line Length" (in tenths of an inch), and in this example, we've set it to "50"(five inches) to make it fit the page size in this book.

Sample letter as it was typed into the Editor

```
;cm SAMPLE/LET for ALLWRITE
;ll 50
;fo off
;in30
March 14, 44 B.C.
;sk2
;in
Julius Ceasar, Imperator
The Senate
Rome
;sk
Dear Julie,
;fo
;PP
It is essential that you stay home from work tomorrow, as the signs are not at all in your favor right now. Many of your supporters have turned against you, and you must take @ *2immediate@ *0 action to protect your safety.
;PP
Since time is of the essence in this matter, I am sending this letter to you by @ $ @ *2EXPRESS SPQR@ % @ *0.
;sk 2;in 30
Your friend,
;sk2
Marc
;in;sk3
@ $Note to Readers, circa 1984 A.D.: @ % This letter was delivered to us just in time to be printed here.
```

Sample Letter as printed by the Text Formatter

Mar 14, 44 B.C.

Julius Ceasar, Imperator  
The Senate  
Rome

Dear Julie,

It is essential that you stay home from work tomorrow, as the signs are not at all in your favor right now. Many of your supporters have turned against you, and you must take **immediate** action to protect your safety.

Since time is of the essence, I am sending this letter to you by **EXPRESS SPQR.**

Your friend,

Marc

**Note to Readers, circa 1984 A.D.** This letter was delivered to us just in time to be printed here.

PRINTING THE LETTER -- <!>

Once you've typed in a letter, you will want to print it. To do so now, turn on your printer, make sure it's at the top of a new page of blank paper, and make sure the printer is "READY." The paper should be positioned so that printing can start at the very top edge, and ALLWRITE will space down to leave a one-inch top margin (we'll show you how to change that later). If you're using "cut sheets" (individual pages instead of continuous forms), position the paper to where the first line of printing should be, rather than to the top edge.

Now, look above "one" on the Soft Key strip. The shifted "1" (an exclamation mark "!"") can be used to select formatted printing. To do so, press CONTROL, then hold down the SHIFT key and press the "one" key. We picked this key because it's easy to use, but unlikely to be hit by accident. The same applies to all the SHIFTED Soft Keys: they do things that will end the editing session. We assigned them that way for safety, but you can change them if you wish to do so. Note that, even though the shifted soft keys issue "commands," you should only press CONTROL and the shifted number. Do not press BREAK, because these soft keys will do it for you.

When you press this particular "FORMAT/PRINT" key, the screen will go blank, and a warning message will appear:

**20 CHANGES NOT YET SAVED TO CURRENT FILE  
REPLY "S" TO SAVE, "Q" TO QUIT ANYWAY, OR  
"-" TO RESUME EDITING**

The number "20" probably will be different in your message.

This message is given for safety. If you had just used a "SAVE" command, the message wouldn't appear, but since some of your changes haven't been saved yet, (the entire document, to be exact), the Editor is warning you of the condition. We could have just taken it for granted that you would always want to save the changes you've made, but anyone who uses a word processor learns quickly that there are times when text should not be saved, so this Editor gives you a choice.

In this case you will want the text to be saved, so press the letter "S". The "ENTER" key should not be pressed, just "S". When you do this, the disks will start to turn, and this message will appear:

**(AUTO) SAVING test**

The actual name will be whatever you chose. Saving a short letter to disk will take only a couple of seconds, and then there will be a short delay while the Text Formatter is brought into memory. Next, your letter will be printed, and when it's finished, the Editor will be restarted automatically.

And that's all there is to it! Unless, of course, you happened to make a mistake, or want to re-word the letter to make it even more effective. If you are the kind of person who types perfectly and never needs to revise anything, then you certainly don't need a word processor; but if you're like me, you probably want to make some changes.

Since the computer is positioned to let you edit the letter again, you can make whatever changes are needed now. If you didn't make any mistakes, you still might want to practice with the "insert" and "delete" control keys, and perhaps add some underlining or boldfacing with the soft keys.

When you've finished your revisions, you might want to run the Text Formatter again. This time, let's do it slightly differently.

**PREVIEWING THE LETTER -- <">, VI Control Word**

We've told you the Formatter has a "preview" capability that uses the screen instead of the printer. Now is the time to use it. Actually, it should have been used the first time, to save the paper, but we thought that, by now, you'd be chomping at the bit to find out just how well ALLWRITE can print.

Pressing <!> (that stands for CONTROL exclamation mark) runs the Formatter using "default" settings. Pressing <"> (Control quotation mark, or capital 2), also runs the Formatter, but lets you override those defaults. Now, press <">. When you do so, you'll be shown the "CHANGES NOT YET SAVED" message again. Reply "S", and after a few seconds, this message will appear:

**ALLWRITE -- FORMATTER  
ENTER FILENAME (DEFAULT = test)**

This means that you could print a difference document than the one you were just editing. That kind of flexibility is characteristic of ALLWRITE, but since we want to use the current file, just press "ENTER" to accept it. When you do so, this message will appear:

**ENTER OPTIONS:**

This is what we have been waiting for, and is the reason we used Soft Key Capital 2 before. "Options" are ways of overriding the built-in defaults. They are entered as a line of control words, starting with a semi-colon and separated by semi-colons. We will refer to them as "run-time options", since they are typed in when the Formatter starts to run. As it happens, we want to use only one option now.

**;VI**

That stands for "VIDEO", and tells the Formatter to send its output to the screen instead of to the printer. After you type it, (including the leading semi-colon), press ENTER. The disks will start to run, and the letter will begin to appear on the screen. If it goes too fast, or you want to examine something carefully, press SHIFT "at-sign" to stop the display, then "ENTER" to get it going again. (SHIFT @ is the standard way to pause things on the TRS-80).

If you're using a 64 character screen, you'll notice that the right hand side of the letter isn't shown. If you want to see it, just press the right arrow, or the shifted right arrow. The text will move sideways (as well as continuing to move up the screen as it is formatted). The left arrow, or shifted left arrow, will move it back. You cannot scroll backwards while previewing text.

**Limitations of Video Output**

The TRS-80 screen only comes in one pitch, and it isn't proportional. So, when you are previewing text that will be printed in proportional, it won't be right-justified on the screen, spaces will tend to be wider than when printed, and the indentation won't always line up. That's just a limitation of the screen, and it won't affect final printing.

**Video Output a Screen At a time**

There's another version of the "VIDEO" option that may be of interest to you, and which you may prefer over "VI":

**;VS**

It stands for "Video Stop," and when you use it, the Formatter will display one screen of text, and then stop. This means you won't have to worry about missing anything. To see the next screen, just press ENTER.

You can switch back and forth between ";VI" and ";VS", by pressing the "CLEAR" key. Scrolling will stop when you do this: press "ENTER" to get it going again or to see the next screen. You can scroll left and right even when the display is paused. In either video mode, you can press SHIFT "at sign" ("@") to pause the display at any time, and then press any other key to resume the output.

**DETECTING FORMATTING ERRORS -- SM Control Word**

ALLWRITE has several features to help you find whether something will print the way you want, without having to actually print it. One of these is the Editor's "VALIDATE" command, which we will cover in the next chapter. The others are Text Formatter features, including the ";VI" and ";VS" control words covered above and "Summary Mode," which we will cover now.

It's helpful to specify Video output when first processing anything over a few pages, since most errors can be spotted easily. If errors are found in any of the control words, the Formatter will switch into a "Summary" mode of operation, and shows only the first line of each remaining page, and any other errors it detects. If output is being printed, this will save a great deal of time, and give you a record of the things to be changed. If you ever want to use Summary mode even when there aren't any errors, you can specify it in the document, or as a run-time option. The control word is:

**;SM (as in "smy")**

and can be used with or without the "Video" output options.

If you try either of the "VIDEO" options, you'll see that a superb representation of the final document is created on the screen: spacing, centering, indentation, page breaks, and emphasis marks are all shown clearly. Because the TRS-80 screen can't show underlining, boldface, and so forth, we've adopted a simple way of showing you where these special effects will occur in the printed document:

- underlining is shown as dashes "---" under the text;
- anything else is shown as tiny squares under the text;
- underlining plus anything else is shown as a thick bar under the text.

An extra screen line is used when necessary to show you the emphasized areas. The results stand out clearly on the screen, and make it easy for you to find out where you forgot to turn something off. By using the "VIDEO" option, you can save an enormous amount of time, since even fancy features such as multiple columns, titles, footnotes, and emphasis marks are shown as they will appear on the paper.

Detecting "pitch" changes (10-pitch is PICA): ten characters to the inch, like most typewriters) on the screen is fairly easy, too: just scroll sideways to the right, so that you can watch the right margin go by. If it changes suddenly, it either means the "Line Length" has been changed or a new pitch has gone into effect.

A word about proportionally spaced text on the screen: obviously, it can't be done on the TRS-80's screen, but ALLWRITE sets up each screen line so that it matches what will be printed later on. The right margin won't be smooth, but the lines will be correct.

While on the subject of previewing, one other important aspect of word processing is finding out where the "page breaks" will occur. When you use ALLWRITE, this usually isn't a problem, because ALLWRITE handles a lot of it automatically, and lets you control the rest in advance. We won't go into the subject now but things like "anti-widowing" and "conditional page" space checking are done for you. When you preview to the screen, the page breaks stand out very clearly, as you may have already seen.

### FINISHING PRINTING

When the Formatter finishes, it'll ask you what you want to do next (it didn't do this in automatic mode):

**PRESS P TO PRINT AGAIN, E TO EDIT, OR Q TO QUIT (DEFAULT=P)**

THE DEFAULT IS "P". After you make your selection, press ENTER. Since we are finished with this letter, just pick "Q" to exit to DOS, or "E" to return to the Editor. "T" sends a "top of form" command to the printer, and is an easy way of ejecting an extra page. However, it is printer-dependant, and not all printers process it correctly.

### END OF FIRST TUTORIAL

This completes the introductory tutorial. ALLWRITE is pretty easy to learn, isn't it? The use of good "mnemonics" makes it easy to remember what keys to press, and the Soft Keys cut way down on the number of keystrokes needed for many things.

The next two chapters cover the rest of ALLWRITE's editing and formatting features. Most of them are just as easy and obvious as the ones you've just learned, but they aren't needed just to write a simple letter. Those chapters are arranged by topic, so when you learn a feature, you'll also learn some others that go with it. The most common topics are presented first, and the most advanced are covered last. Whether you decide to keep reading now, or to just dive in and rely on the index and HELP key to bail you out, is up to you. Either way, you have probably learned enough for most common purposes, and you will learn more as you need to. Thank you for staying with us so far, and we'll continue in the next chapter.



**CHAPTER 4**  
**ADVANCED TUTORIAL – EDITOR**

This chapter will teach you how to use the other features of the Text Editor. Features covered in Chapter 3 will not be repeated unless they have extended capabilities not already covered. Although the topics are presented in tutorial fashion, we don't expect you to just read through the chapter: after reviewing the outline below, you can just look things up as you need them.

**TOPICS IN THIS CHAPTER**

Warm Restart:	type "AL" to recover text in memory
Quick Start:	type file name after "AL"
Blocks:	move, copy, delete, put, list
Search and Replace:	search, SU, flow, case, <F>, </>, replace, <R>, change
Named Points:	up to 26 at a time
Tabbing:	on screen and/or at print time

Miscellaneous Control Keys:

screen print	splint a line
blank lines	joining lines
word reversal	on-screen underline
hard space	special symbols
replicate character	shift line up

Miscellaneous Commands:

line joining	screen width
autosave	printer selection
repeat last command	display last command
validate emphasis marks	

File Commands:

save	kill
name	end
new	run
options	area

Long Documents:

link	append/imbed
------	--------------

Soft Keys:

defining	saving
loading	

**RECOVERING TEXT AFTER A SYSTEM FAILURE**

The words used by computer people for a system failure are "crash", or "hang up." Actually, those aren't the words they use when it happens to them, and we won't repeat what they say here. A system failure is a fancy way of saying that something went wrong, really wrong, with the equipment or the software, and nothing works anymore. This can be caused by a momentary power failure or surge (turning on a washing machine or a refrigerator is a wonderful way to demonstrate this on the TRS-80 Model 1. Also, an intermittent circuit problem in the computer, or a "bug" (a programming error) in DOS or ALLWRIGHT (impossible, of course).

Now, as everyone who has used a computer for a while already knows, it will never crash when you are writhing the first page of a paper that is due in three months. Computers may not be able to "think", but they certainly seem to know about deadlines. If a computer is going to crash, it will do so the night before your paper is due, probably when you're writing the last paragraph, and are so tired that you've forgotten to "SAVE" the text for the last five hours. Then it will crash, or if it's feeling really diabolical, it may decide to "hang", instead of "rebooting" spontaneously.

If (when ?) this happens to you, don't panic, because ALLWRIGHT has an excellent, easy-to-use way of recovering the text that's still sitting in the memory of your computer. This is what you should do:

1. DO NOT TURN OFF THE COMPUTER. If you do, or if there's been a power failure, the text cannot be recovered.
2. Press the "RESET" button on the computer while holding down the "ENTER" key to prevent any "AUTO" command from occurring. If you know that the "AUTO" command won't harm the text area, it's O.K. to let it occur; if you aren't sure, or don't know what it will do, hold down "ENTER" until the "DOS READY" message for your Operating System appears on the screen.

Type: **AL \***

3. and press the "ENTER" key. The asterisk "\*" tells ALLWRITE to attempt a "warm restart", which means that it will try to recover and use whatever it finds in memory. If you omit the asterisk, a normal "cold start" will occur, the text will be lost for sure, and it won't be possible to try a second time.

Note: If you are using Model 1 TRSDOS 2.3 or NEWDOS 2.1, do this instead, since the "BREAK" key doesn't work with those DOS's unless "ALK" is in use:

**ALK AL \***

4. The Editor will start and then display on the screen what ever it finds in memory. If what it displays looks good, then you should "scroll" through the entire text to make sure it's O.K. If so, save it to disk, but under another name. We haven't covered that yet, but the command (press "BREAK" to enter a command) is:

**CMD=> SAVE TEMP/TXT**

5. After the "SAVE" is done, press the reset button again, start ALLWRITE normally (without the asterisk), and when asked for the document I.D., reply "TEMP/TXT". Examine it again, and if it still looks O.K., change its name to whatever name you had been using, and continue editing it. The "NAME" command is used for this:

**CMD=>NAME BIGPAPER/CH9 (pick any valid name)**

This recovery procedure avoided using "ALK", as well as any other special routines you might normally have in your computer. We can't predict what will cause your problem, but you will want to get that text saved to disk, under a temporary name, as quickly as possible. The fancy stuff, including figuring out what went wrong, can come later.

**QUICK START**

When you type "AL" from DOS, you can include the name of the file to be edited:

**AL SAMPLE/LET**

If you do that, ALLWRITE will read that file into memory without prompting you for a document I.D. If you are using the "ALK" keyboard driver, which by default activates "AL" also, you can still do a quick start.

**ALK AL SAMPLE/LET**

As you can see from this example, you will have to type "AL" as well as the name of your file, so you may not want to do it this way.

When you start with "ALK", you don't have to let it activate "AL" as well: you can type any command or program name after "ALK", and that command will be run. If you want to use "ALK" as a keyboard driver, but not run anything immediately, you can follow "ALK" with a dash: after the keyboard driver is activated, it will return you to DOS:

**ALK -**

**BLOCK OPERATIONS**

A "block" is a section of text in a file. It may be as short as a single character, or as long as the entire file. The "delete" control keys work with specially defined blocks: a character, a word, a line, a sentence, or a paragraph. However, you may want to manipulate an area of text that starts and ends at any points whatsoever. To do that, control keys are needed to let you define your own blocks. These blocks can encompass lines, sentences, and paragraphs, and can start and end in the middle of anything.

This kind of block definition is discarded as soon as the block operation finishes, so you can not accidentally destroy it later on. If you want to keep re-using a block, you can set "points" to identify it. The use of "Points" will be explained later.

These are things you can do with a "block":

- move it to another place in the file
- copy (duplicate) it to another place
- delete it
- list it to the printer (not formatted)
- write it out to disk (put it to disk)

There's a command called "GET" that does the opposite of "put", and it will be covered also.

**Marking The Beginning of A Block -- <B>**

To manipulate a block, you must identify where it starts, where it ends, and possibly where it is to go. To tell the Editor where a block starts, just move the cursor to the first character of the block, then press <B> (Control B). When you do so, the cursor shape will change to a graphic that looks like a lower-case "b".

While the "block" cursor is on the screen, the Editor will not allow you to make any changes whatsoever to the text: you can't overstrike, insert, or delete anything (except the block itself). This restriction is necessary because changes might shift the text around, causing the Editor to lose track of where the block starts or ends. Attempts to make changes are just ignored when block mode is in effect.

To tell the Editor where the block ends, move the cursor to the last character of the block (not the character after the end of the block, but the last one within it), press the control key again, and then one of the keys shown below (remember, the brackets just stand for the control key).

- <C> to COPY (duplicate) the block
- <D> to DELETE the block
- <L> to LIST the block to the printer
- <M> to MOVE the block
- <P> to PUT the block in any disk file

If it's more convenient, you can define the bottom of the block first with <B>, and then define the top of the block with one of the keys listed above. The results will be the same.

Note: <D>, <L>, and <P> are block control keys when used after <B>, but have other functions when used by themselves. If you make a mistake, you'll get a warning message and have a chance to cancel the incorrect keystroke.

#### **Block Copy <C> and Move <M> -- to "HERE" <H>**

If you choose <C> or <M>, you'll also have to tell the Editor where the block should go. To do so, move the cursor to the character that is the target of the move or copy, then press <H>, which stands for "HERE". If the operation is "COPY", a duplicate of the block will be placed just before the character marked by <H>; if the operation is "MOVE", the block will be deleted from its old place and inserted just before <H>. This is sometimes called a "cut" and "paste" operation, but in ALLWRITE, the block to be moved is never left aside where it can be lost if you forget to do something with it. Please note that <H> cannot lie between the start and end of the block. If it does, this error message will be displayed and the block operation will be canceled:

#### **TARGET CAN'T BE INSIDE BLOCK**

#### **Moving or Copying a Single line -- <M>,<C>**

If you want to move or copy just one screen line at a time, there's a shortcut available: move the cursor onto the line (any place on the line is O.K., you don't have to be at the start or end), and instead of pressing <B> to begin a block operation, just press <M> or <C>. If you do this, ALLWRITE will define the start and end of the line containing the cursor as the block to be moved or copied. Then, all you have to do is move the cursor to the target and press <H>, just as before.

#### **Block Delete -- <D>**

If you choose <D> (delete), you will be asked to confirm that you really mean it. This is similar to sentence and paragraph deletion, so the Editor will try to help you avoid the consequences of an erroneous keystroke. Once you confirm that the deletion should occur, "WHOOPS" can't be used to recover the deleted text. If it's still on disk because it had been saved there previously, it can be recovered by one of the "File" commands that will be explained later.

#### **Block List (printing as-is) -- <L>**

If you choose <L> ("List," similar to the "LLIST" in BASIC), the block will be printed immediately, as-is, with no text formatting. This is useful when you want an immediate record of what you've written, when you're trying to figure out why something isn't printing properly, or when you are reporting a problem to us (obviously, that third reason couldn't ever, possibly, happen). The width of the printed lines will be the same as the current line length assigned to your screen: we will explain how to change the line length later on. Note that Block List is different from a "screen print", since it can show you text that isn't on the screen right now. To print an image of what's on the screen, simply use <\*>, without any <B>.

You can cancel the printing operation at any time by pressing the "BREAK" key, even if the printer isn't ready. However, if you're using a Model 4, it may take a few seconds before DOS allows ALLWRITE to continue. If you are not using our "printer driver" support (a choice you made during installation), pressing "BREAK" will not work if the printer is in a "not-ready" state. If this happens, you must either "Ready" the printer to continue; or else press the "RESET" button on your computer to clear the DOS problem, and then use ALLWRITE's "warm restart", as explained at the beginning of this chapter.

#### **Block Putfile - - <P>**

The "Block Copy" and "Move" operations we just covered can only move text from one place to another within the same in-memory file, and the amount of text you can store in a single file is limited by the size of memory in the computer. There will be times when you will want to use a block of text that is in another file, and there are several ways to do this.

The first of these is "Block Putfile." It lets you define a block, and then to make a copy of it on disk. The original block is not deleted from the in-memory file (you can delete it afterwards). You can put (write) the block to a "new" disk file, replace an old one, or add (append) the block to the end of an existing one.

The end-points of the block are defined by <B> followed by <P>. <P>, by itself, means "paragraph delete", but if <B> was pressed first, the Editor understands that "Putfile" is being requested. If you forget to press <B> first, just press "ENTER" to cancel the "paragraph delete", and then use <B> to begin the "PUT" correctly. When you press <P>, you will be shown this message:

**ENTER I.D. TO BE USED FOR OUTPUT FILE,  
OR QUESTION MARK "?" AND ENTER TO READ A DIRECTORY,  
OR MINUS SIGN "-"(AND ENTER) TO RETURN TO EDITING**

Reply by typing the name of the file that is to store the block of text. If it is a "new" file, the block will be written to it, then the text screen will be redisplayed. If the file already exists, the block will not be written to it. Instead this message will be displayed:

**\*\*\* FILE ALREADY EXISTS \*\*\*  
PRESS <ENTER> TO APPEND TEXT TO FILE,  
OR "/" (AND ENTER) TO REPLACE IT OR**

**ENTER I.D. TO BE USED FOR OUTPUT FILE,  
OR QUESTION MARK "?" AND ENTER TO READ A DIRECTORY  
OR MINUS SIGN "-"(AND ENTER) TO RETURN TO EDITING**

The default is to append the block of text to the end of the existing file. This is useful when extracting several blocks of text from one file for inclusion in another file. However, you can replace the file or choose another name, instead.

Once you've supplied a name to "PUTFILE", it will remember it for the remainder of the editing session. So, if you use <B> and <P> again, the above messages will appear, followed by:

**DEFAULT IS: name of file**

That is, ALLWRITE will assume you want to keep appending blocks of text to the same file, but it will let you change files or cancel each time. To confirm the "append", just press "ENTER". This method of operation is quick, requires very few keystrokes, and is safe: the block is not deleted from the file you're editing.

**COPYING TEXT FROM DISK INTO MEMORY - - GETFILE Command**

If you use the "PUTFILE" control key to write text from memory to disk, you are likely to want a way to copy part of another file into the one being edited. That is done by a command called "GETFILE", and it's flexibility allows you to be selective in what is copied. Besides just copying a small block of text from disk to memory, it can be used to edit portions of a file that is too large to fit in memory; we'll cover that sort of thing separately, later on.

To use "GET", move the cursor to the character just after the position the text should be placed (just like an "insert" or Block Move), and then press "BREAK". Next type "GET", the name of the file to be used, and possibly the range of text to be included. Finally, press "ENTER" to execute the operation. The command looks like this:

**CMD=> GET filespec [/from/[thru/]]**

For example      **get contracts /termination/conditions./**

To explain the command, we will have to explain what that line really means.

GET is the command (press BREAK first). It can be abbreviated to "GE" since two letters are always enough to uniquely identify commands. At least one space must follow "GE". It is shown capitalized to indicate it must be typed as shown.

filespec stands for the name of the file from which text is to be copied. It is shown in lowercase to indicate that you must supply a name of your own; in general, you won't be calling your letters "filespec", but this notation reminds you to make a substitution.

[] indicates that what is within the square brackets is optional, and may be omitted if you don't need it; the brackets themselves should not be typed as part of the command line. If you press ENTER right after typing the "filespec", the entire file will be copied into memory.

/from/ is one of the optional "parameters" that can be used with "GETFILE". The slash "/" is a "delimiter", and it marks the start of the word or words ALLWRITE should look for when it reads through the file. ALLWRITE doesn't look for the delimiter character, but only for what lies between each pair of delimiters. In fact, the delimiter cannot be any of the characters for which the Editor will be searching, and nothing will be copied from the file until and unless "from" is found. If it isn't found, nothing will be copied, and this message will appear:

#### **TEXT NOT FOUND**

If "from" is found, all text, including "from", will be copied into memory. This will continue until the end of the file is reached, or until you run out of room in memory, or until "thru" is found. If you don't include "thru" (which will be another word or words), then the rest of the file will be copied into memory.

thru is also optional. It's used to indicate where copying should stop, and the text including "thru" will be copied into memory. Note that the delimiter (the slash, in this case) separates "from" and "thru" and also ends each of them. This means you can tell ALLWRITE to look for things that include blanks.

**EXAMPLES:**

```
get jerry/let
ge jerry/let /last week/prices
ge jerry/let *last we*
ge jerry/let //prices/
```

In the first example, the entire "JERRY/LET" file will be copied into memory and become part of the text you're developing now. In the second example, only the block of text in "JERRY/LET" beginning "last week" and ending "prices" will be included. In the third example, the block beginning "last we" and ending at the end of the file will be included. In the fourth example, text will be copied starting at the beginning of the file (there's nothing, not even a space, between the first two delimiters) and ending at "prices". In all cases, "JERRY/LET" will remain intact and unchanged on disk.

These examples also illustrate that the "delimiter" doesn't have to be a "slash;" it can be any character that doesn't occur within "from" or "thru". Also, regardless of what you choose as a delimiter, you must use that same character throughout the command. You can pick a different one next time, but all three occurrences of the delimiter must be the same character. For example, the following wouldn't have worked in example two above:

**BAD EXAMPLE:**      **ge jerry/let**      **?last week#prices./**

ALLWRITE would scan the file, looking for: "last week#prices./", since the question mark is the delimiter in this case. And, since it probably wouldn't find anything like that, nothing would be copied into memory.

### Copying Text from One File To Another

We've really been talking about this as we covered "PUT" and "GET". The steps are:

1. edit the file that already contains the text;
2. use "BLOCK PUT" to store a copy of the text into a new, temporary disk file;
3. if the text is no longer needed in the current file, use "BLOCK DELETE" to get rid of it;
4. "SAVE" the file if you've made any changes to it (such as deleting that block);
5. edit the file into which the block is to be transferred. We haven't covered ways of switching from one file to another yet, but you can use the "NEW" command: when asked for the name of the file to be edited, reply with the name of the other file;
6. scroll through the second file until you find where the extracted block of text is to go, and position the cursor there;
7. use "GET" to include the temporary file;
8. check the text to make sure it's complete, correct, and where you want it to be, then "SAVE" the updated file;
9. if you want to remove the temporary file from the disk, you can use the Editor's "KILL" command. We haven't covered it yet, but it works just like the DOS command. On the Model 4, the DOS command is "REMOVE", but ALLWRITE just uses "KILL".

This completes the material on "BLOCKS" and "Inter-file Transfers."

### SEARCH AND REPLACE

This is one of the most important topics in any discussion of Word Processing, because it lets you use the computer to do things that are just about impossible with pencil and paper. Everything we've covered until now could have been done with pencil, paper, an eraser, and a pair of scissors; but hunting through ten pages of text to correct a repeated spelling error is not only very slow and tedious, but also prone to error: you just know you'll miss some of the occurrences of the word, especially if it's important and you're in a hurry.

On the other hand, a computer really shines when it comes to repetitive tasks that would require endless patience and attention to detail. It may not be able to think, and its creativity may be limited to mangling your prose when there's an equipment or software failure; but when it comes to finding every "precious" and changing all of them to "precious", a computer becomes a thing of joy and wonder.

Let's define the terms we'll be using: "searching" means scanning through text and trying to find something that matches something else. "Replacing" means deleting something that's already there, and inserting something else in its place. An "argument" is a series of characters that is to be found or replaced in the text.

Because Search and Replace capabilities are so important in word processing, ALLWRITE has several ways of doing them, including some shortcuts to reduce the number of keys you must press. Let's start with "searching."

### **The SEARCH Command**

To tell ALLWRITE to search for something in the file currently being edited, press the "BREAK" key, type "SEARCH", one space, and whatever you're looking for. Then, press "ENTER". The thing you're looking for is called a "Search Argument," and if it's found, ALLWRITE will move the cursor to it. If it wasn't on the screen, the text will be scrolled until the matching text is at the top of the screen.

When the command is used as described above, ALLWRITE makes a couple of assumptions on your behalf:

1. the search begins at the first character after the cursor, and continues forward through the file until a match is found or the end of the file is reached;
2. upper and lower case are both acceptable, and in any combination. If you were looking for "Smith", for example, ALLWRITE would accept "SMITH", sMITH", and even the last five letters of "blacksmith." There are ways to tell ALLWRITE not to be so open-minded, and they will be covered soon.
3. if you're looking for something that involves more than one word, it may happen to be split across two screen lines, but ALLWRITE still will find it, unless the first of these lines ends with an "ENTER" symbol. If you want "search" to ignore "ENTER" symbols and treat them as blanks, you can use the "FLOW" command to change the way "ENTER" is handled by "search."

Like all other commands in ALLWRITE, "SEARCH" can be abbreviated. Its official name is just "SE", but "S" alone will do. In both cases, one space must follow the command, to let ALLWRITE see where the search argument begins.

If what you're looking for has spaces before or after it, you can type those in too. The first space doesn't count, since it just ends the "SEARCH" command, but all spaces after that are treated as part of the search argument.

To avoid an ambiguous situation such as the "blacksmith" example above, you can often include leading and trailing blanks in the search. "smith" occurs in "blacksmith", but " smith " does not.

Because searching is used so much, there are three synonyms for it: "FIND" (abbreviated to "F"), "LOCATE" (abbreviated to "L"), and the slash symbol, "/". They all do exactly the same thing, and we've provided synonyms so that you can pick whatever word seems most natural to you. Another reason for having synonyms will become apparent when we show you the control keys that can be used to repeat a search. When "F" is used, it must be followed by a space, but when "/" is used, no space follows it. That's one of the shortcuts, and is also a method familiar to those of you who have used either NEWSCRIPT or an IBM mainframe editor in the past. For example, the following commands all do the same thing:

```
search woody
Se wooDy
s WOoDy
/woody
F woody
Loc woody
```

The "SEARCH" command is fairly fast: It scans a 25,000 character file in four seconds on a Model I, or two seconds on a Model 4.

If the search argument is found in the file, this message will appear, but no harm will have been done.:

#### TEXT NOT FOUND

If what you needed happened to lie "above" the cursor, the search wouldn't find it either, or might find another occurrence of it. There are two things you can do about this:

1. start from the top of the file: <T>
2. use the "SEARCH UP" command: SU

#### Searching Backwards -- "SU" Command

Just as "SEARCH" runs towards the end of the file, SU ("Search Up") runs towards the top of the file. The search starts on the character just before the cursor, and the search argument is defined just as it was for "SEARCH". If something is found, the cursor will be positioned at the start, not the end, of the match.

#### Approximate (Masked) SEARCH

An "approximate" is one in which certain character positions don't count: if the rest of the text matches the rest of the search argument, the search will succeed. This is also called a "masked" or "wildcard" search. For example, if you were willing to accept either "report" or "resort", you wouldn't want the third character to be checked.

To select an approximate search when you type the "search argument", use the ASCII character "255" in positions that should not be checked. This character can be entered as a "Special Symbol", a topic we will cover later on.

Please note that the ASCII "255" is reserved for this use, so you cannot search for it. However, you can use it in normal text. Also note that the "approximate" character can be used as the search argument in "Replace" and "Change", which are commands we will discuss soon.

#### Repeating a SEARCH -- <F>

(This control key can only be used after a "SEARCH" command has been issued.) It turns out that searching for a specific thing often has to be done over and over again; this is just one of the characteristics of revising text. If you had to re-type the "search argument" each time, it would be better than hunting by eye, but annoying. Now, we wouldn't have brought this up if we didn't have a solution: ALLWRITE "remembers" the last "search argument" until you define a new one, or until you exit from the Editor. It even remembers the search argument when you switch editing from one file to another.

What makes repeated searching really easy is that it can be done with control keys: once you've used a search command to define a search argument, you can repeat the search by pressing <F> or </> (that's why they are synonyms for "Search"). The search will start at the character just after the cursor. If "SU"(Search Up) was on the last search command used, the search done by the control key will also scan backwards.

To remain compatible with earlier word processors, "LOCATE" and "LU" are also recognized by ALLWRITE. "LOCATE" and "FIND" are the same as "SEARCH" and "LU" is the same as "SU". So, if you're new to this sort of thing, there's no reason why you should learn more than one of these command words: we provided synonyms for convenience only.

### REPLACING TEXT

There are four ways to replace text while editing with ALLWRITE:

- type over it (overlay/insert/delete)
- the "REPLACE" command and control key
- the "SR" (Search and Replace) command
- the "CHANGE" command

The first of these methods uses the simple editing features covered in chapter three. It's the best way to make one-time changes when the text is right on the screen.

#### The "REPLACE" Command

The "REPLACE" command (abbreviation: "R") is typed in just like the "SEARCH" command: one blank must follow the command itself, and everything thereafter becomes part of the "replacement argument," including leading and trailing spaces. "REPLACE" can't be used until a "search argument" has been defined, and even then, it will succeed only when the cursor is positioned to the start of text that matches the "search argument." This may sound complicated, but all it means is that you must do a "search" just before you do a "replace."

Once a "replacement argument" has been defined, ALLWRITE will remember it, just as it remembers a "search argument." So, you can keep using it over and over, even if you switch to editing another file. These arguments are remembered until the Editor is terminated or new arguments are defined.

This method was chosen because it is very safe, and because, in certain conditions, is very convenient. It is safe because text can be replaced only when ALLWRITE determines that the "old" text matches the "search argument". If the text at the cursor does not match the "search argument", then the "replacement argument" still will be defined, but this message will appear and no changes will be made:

#### TEXT NOT FOUND

The message refers to the fact that the "search argument" doesn't match the text that begins at the cursor; it has nothing to do with the contents of the "replacement argument".

#### Repeating a REPLACE -- <R>

A control key can be used to repeat the latest "Replace". The key is <R>. It will work only if a search argument and a replacement argument both have been defined previously, and if the cursor is positioned to the start of an occurrence of the search argument. When these conditions are met, pressing <R> immediately changes the text in question, and advances the cursor so that it's positioned just after the end of the text it replaced.

The reason this Search/Replace method is convenient now becomes clear: <F> can be used to perform a Search, while <R>, directly above <F> on the keyboard, can be used to perform the corresponding Replace. You can keep holding down the Control key while pressing <F> to move the cursor to the next occurrence of the "search argument". Then, if you want to replace that occurrence, just press <R>. If you want to leave this particular occurrence alone, press <F> again to continue the search. This is a safe way of doing selective replacement.

By the way, if "SU" (Search Up) had been used to define the search argument, <F> and <R> would proceed backwards through the text, eventually reaching the top of the file.

**Automatic Search and Replace -- "SR" Command**

Pressing <F> and <R> alternately is an excellent method in many cases, but if you had to replace something many, many times, and didn't have to be selective about it, it would be a lot easier if you could just tell ALLWRITE how many times to do it for you. Needless to say, there is a way, in fact a couple of ways, to do this.

The "SR" command does a "Search", and if a match is found, it does a "Replace", all in one step. It only goes forward, towards the end of the file, even if "SU" had been used. Before using it, you must have used both "Search" and Replace" to define the necessary "arguments". If that was all there was to "SR", it wouldn't be worth mentioning; but if you put a number after it, the search-replace will be repeated that number of times. For example:

```
CMD=> /the
CMD=> r this
CMD=> sr 47
```

will replace the next 47 occurrences of "the" by "this". If you don't give a number, "1" is assumed. If you give too many, no harm will be done (it stops at the end of the file). If you want to replace all the remaining occurrences of something (from the cursor to the end of the file), you can specify an asterisk "\*" instead of using a big number. That's called "global search and replace":

```
CMD=> SR *
```

Before using it, the "SEARCH" command and the "REPLACE" command both had to have been used. Once set, those "search" and "replace" arguments remain defined when you link from one file to another, so you can repeat "SR \*" to change every file in a chain, quickly, easily, and safely.

For safety, we have not provided a fully automatic way of making global changes across several files. However, at the end of the "Soft Key" topic, which is at the end of this chapter, we will show you how to set up a semi-automatic way that is safe, but uses only three keystrokes per file. The heading to look for is "A Soft Key For Cross-file Changes."

"SR" doesn't move the cursor, so when it's done, the screen and cursor will be positioned just as they were before.

**Number of Changes Made**

When "SR" finishes, it will display a message that shows the number of occurrences it found and replaced. After you've read the message, press "ENTER" to re-display the text screen.

**The "CHANGE" Command**

The "SEARCH" and "REPLACE" method described above has several advantages and one drawback. The advantages are: 1)if you use the <F> and <R> control keys, you can examine each occurrence before changing it, thereby avoiding a costly error; and 2) ALLWRITE stores the arguments so that you can easily re-use them later on, even if you've switched to editing another file.

"SR" doesn't move the cursor, so when it's done, the screen and cursor will be positioned just as they were before.

The drawback is that you have to use at least three separate commands to get a multiple or global search and replace. To avoid this inconvenience, ALLWRITE also has a "CHANGE" command. It lets you give the "search argument", the "replacement argument", and the number of times it should be done, all at once. The command format is similar to the one used for "GETFILE":

**CMD=> C /old/new/ n**

"C" is the minimum abbreviation for "CHANGE". The slash "/" is a "delimiter", and may be any symbol that doesn't occur in either "old" or "new". "old" is the search argument, "new" is the replacement argument, and "n" is the number of times the command should be repeated. The default is "1", and if you want a "global change", you can use an asterisk instead of a big number. When the command finishes, it'll display a message showing how many occurrences were found and replaced.

**Example:**      **CMD=> c ?Bill?William? \***

This command is fairly fast: if you had a 20,000 character file, you could change all the "e"s to "\*\*\*" in less than 5 seconds. (But, if the editor was ignoring upper/lower case, the capital "E"s would be changed too.) The big advantages of "CHANGE" are that it's a great shortcut and it's very fast. The drawbacks are that ALLWRITE doesn't "remember" the arguments for next time (but you can assign the command to something called a "soft key", which we will describe later on), and that you can't check each change before it takes place (if you need to, you can use <F> and <R>).

"CHANGE" doesn't move the cursor, so when it's done, the screen and cursor will be positioned just as they were before.

If you aren't used to this kind of editing command, we suggest that you "SAVE" your text before doing a "global change", and check the results over before doing another "SAVE". "Changes" can't be cancelled except by not saving the file. There's a tradeoff between speed and safety, so when you want the safer approach, use <F> and <R>.

### **HOW CHANGES ARE COUNTED**

Each "Replace" or <R> counts as one change. Each "SR" and each "CHANGE" counts as one change, even if a repetition number is given: "SR 5" counts as a single change. This method of counting was chosen to avoid triggering "AUTOSAVE" in what could be a dangerous situation: if each occurrence were counted as a separate change, and "AUTOSAVE" were set to "50", a global change would very likely trigger the automatic saving of text to disk. Since we just finished explaining potential dangers of global change, incrementing the count only by one reduces the likelihood that an incorrect global change would be saved to disk, thereby destroying a good copy of text.

When you use <N> or <ENTER> to open a new line for typing, it counts as a change. When you actually type on that line, it counts as a second change.

### **COUNTING OCCURRENCES OF TEXT**

To find out how often a particular word or phrase occurs in your text, just change it to itself, globally, starting at the top of the file:

**CMD=> c/ house / house / \***

(there is a space before and after each "house".) When "CHANGE" finishes, it will display the number of occurrences.

**UPPER AND LOWER CASE -- THE CASE COMMAND**

By default, ALLWRITE's "Search" treats upper and lower case letters as being identical: "A" and "a" would match during a search. That's usually good, since the word you're looking for might be the start of a sentence. But, if you told ALLWRITE to:

**CMD=> change /this/that/\***

Then, whenever "This" was found at the start of a sentence, it would be replaced by "That", and the sentence no longer would begin with an upper-case letter. You couldn't get around the problem by using:

**CMD=> c/this/that/\***

because the occurrences of "this" in the middle of sentences would be changed to "That" as well.

The solution, of course, is to tell ALLWRITE when to pay attention to "case":

**CMD=> CASE YES (or) NO**

When ALLWRITE starts running, "CASE NO" is in effect, and upper / lower case are treated alike. If you specify "CASE YES", then "T" and "t" will be treated as different symbols. This will remain in effect until "CASE NO" is issued, or until the editor is terminated.

By the way, "ON" or "1" are synonyms for "YES", while "OFF" and "0" (zero) are synonyms for "NO". So, if "CASE OFF" makes more sense to you, you can use it instead.

**SEARCHING FOR THE "ENTER" SYMBOL -- The "FLOW" Command**

As we explained earlier, when searching, ALLWRITE normally treats "ENTER" (which is really a "Carriage Return") as an actual character that is different from a "space". You will recall that "ENTER" appears on the screen as a backward "L". Usually, you will want the Editor to distinguish between "ENTER" and a space, but if you ever want a "search" to treat "ENTER" symbols as spaces (blanks), you can use this command:

**CMD=> FLOW ON (or) OFF**

The command is called "FLOW" because of how it affects the view of text during a search. "ON" tells "SEARCH" to treat "ENTER" symbols as though they were blanks, while "OFF" restores the normal distinction between them. This command has no effect on the text itself, nor on what you will see on the screen. If you don't specify "OFF", "ON" will be assumed when the command is issued.

**NAMED POINTS**

The last feature in the "search and replace" topic is different from the others. Rather than searching through the text for a particular "argument", this feature is used to tell ALLWRITE to remember where the cursor is right now, so that you can come back to the same point later on. It's very useful in "Block" operations, when you may want to copy the same piece of text over and over to different places in the file. It's also useful when you want to see just what you had written about a subject so that you can remain consistent later on.

To tell ALLWRITE to record where the cursor is right now, press the control key, then the comma ",", and then a letter (A-Z). Upper and lower case are identical, so you can have up to 26 named points at one time. Note that this is a three-keystroke sequence.

To "recall" a named point later on, press the control key, then the period "." (for point), and then the letter that names the point in question. When you do this, one of three things will happen:

1. If the name has not already been defined with "<,>", the request will be ignored, and no harm will be done;
2. The text will be re-positioned onto the screen, with the cursor in the same line and column as it was when the point was defined. If no text insertion or deletion has occurred "above" the point since it was defined, the screen will look identical to the way it had been at the moment of definition;
3. The text will be re-positioned onto the screen, but if text deletion or insertion "above" this screen has occurred since the point was defined, the text may be positioned differently, and the cursor may not be placed on the same character as it was originally.

This completes the discussion of the "Search and Replace" group of commands and control keys.

## TABBING

Tabs are used to position text and numbers into columns. ALLWRITE has excellent tabbing and columnar capabilities, including the ability to print text in multiple-column format, similar to that of a newspaper; but that isn't what we're discussing right now. There are two kinds of tabbing facilities available: 1)on-screen, which is done while editing to let you see where things are; and 2) print-time, which is more flexible than on screen, but does not show the positioning until printing or previewing occurs.

Print-time formatting is the only practical means of using tabs with proportional printing or decimal alignment of numeric columns. It is also the preferred method when features such as "hanging indents" are used. On-screen formatting is the preferred method when laying out tables that will be printed "as-is" in non-proportional pitch. This section will explain how to use both methods, beginning with on-screen tabbing.

### On-Screen Tabbing

On-screen tabbing is done by using control keys. Tab definitions can be saved onto a disk and then loaded for use afterwards: this is done with the "TAB" command. The control keys are as follows:

<right arrow>	move cursor right to next tab stop does not wrap around to next line
<left arrow>	move cursor left to previous tab stop Does not wrap around to previous line
<@>	move cursor right to next word
<G>	display or remove "grid" line showing tabs several grids can be on-screen at once

The command is:

TAB SAVE filename	tabs are saved to "filename"
TAB LOAD filename	new tabs are read from "filename"

### Initial Tab Settings

The initial tab stops are set every 8 positions 8, 16, 24, 32, etc.

### Tab Keys

Control right arrow is just like the tab key on a typewriter, and Control left arrow is the same thing, but it lets you move to the left (unlike most typewriters). Control at-sign <@> tabs forward a word at a time: when used in conjunction with the left and right tab keys, it offers a very fast way of moving the cursor around.

### The "Grid Line" -- <G>

The "grid line" is a guide to show you the column positions on the screen and where the tabs are set at the moment. Initially, It looks like this (not all of it is shown here):

```
----:+0---:+-20:---+--30:---+--40+----:+-0:---:+-60:---+--70:---+--80
```

(The grid can appear on any line except the first, second, and last.) Each dash "-" marks a screen column position; each colon ":" marks a multiple of five; every tenth column is numbered just to the left of the colon marking that column; and each tab is marked by a plus sign "+" that takes precedence over anything else in that position of the grid. That's why the number "10" isn't shown in the example above: there's a tab set in column "8", which is where the "1" of "10" would otherwise have been shown.

Although the tab keys work whether or not the grid line is displayed, it often helps to have it on the screen as a guide, particularly if you're working with a line that is wider than the screen. If this is the case, the grid line, as well as the text, can extend past both sides of the "window" represented by the screen.

To display a grid line, just press <G>. To remove one from the display, move the cursor onto the grid line and press <G>; it acts like a toggle switch. You can have several grid lines on the screen at once: one near the top and one near the bottom, for example. Remember: the grid cannot be displayed on the first, second, or last lines of the screen, and all grid lines will be identical (only one set of on-screen tabs at a time, although you can change them to lay out different tables.)

### Setting and Clearing Tabs

Tabs can be set and cleared only when the cursor is on a grid line. The cursor can be moved left or right with the left/right arrows (normal or control), but not by the space bar.

To set a tab, position the cursor where you want the tab, and press the plus "+" sign or the semi-colon. When you do this, a "+" will appear as confirmation of the setting, and the cursor will move one position to the right.

To clear a tab, position the cursor over a "+" and press the minus "-" sign. The plus sign will be replaced by a dash, and the cursor will move one position to the right.

When the cursor reaches the extreme right side of the grid, it just stops if you were tabbing across, or moves to the next line if you were using the right arrow. Once you've finished setting your tabs, you can leave the grid on the screen or remove it, as you prefer. The tabs still will work.

No facility for clearing all the tabs at once has been provided, in an effort to make it a little harder for you to lose all the settings by mistake.

Unless stored on disk (see below), tab settings are lost when ALLWRITE is terminated. However, you can edit several files in a row, print them, use Electric Webster or Dotwriter, and not lose the tab settings.

### Saving and Loading Tab Definitions on Disk

If you have several standard tab setups, you can define each of them once, and store them on disk for future use. Each setup must use its own small file. The tabs should be set by the method described above, and once all tabs for one setup are complete, this command should be used:

**CMD=> TAB SAVE fileid**

"TAB" can be abbreviated to "T", "SAVE" can be abbreviated to "S", and "fileid" is whatever name you choose for this particular tab file. (Try to pick something that's descriptive of the setup or report it goes with.)

To start using a previously saved tab setup:

**CMD=> TAB LOAD fileid**

This is similar to "SAVE", but "LOAD" can be abbreviated to "L", and "fileid" should contain valid tab settings. If you use a file intended for some other purpose, no harm will be done, but the tab positions will be really strange. If "fileid" isn't found, you'll be shown this error message:

**FILE NOT FOUND**

That's all there is to "on-screen" tabbing. It's easy to understand and very natural to use. However, it has limited flexibility: The screen can't show proportional spacing; the editor cannot do fancy tabbing, such as decimal alignment; and since the text is stored in fixed positions, if you decide to change your tab columns, you'll have to move all your text around to fit the new requirements. For those reasons, it often will be much more satisfactory to use the text formatter's "control word" for tabbing. That's covered next.

### Print Time Tabbing with Control Words

In the first tutorial, we introduced the use of "Control Words" as a means of specifying print format requirements. ";CE" is used to center text, for example. The control word used to describe tab stops is ";TB" (as you might already have anticipated). It must be followed by the symbol you will be using to indicate tab stops, and then by definitions of the tab positions. To be consistent with other horizontal measurements, tab positions normally are expressed in tenths of an inch, but there's a way to tell ALLWRITE to use character columns when in 12 or 16 pitch. We'll explain that later on, but will describe tabbing in terms of tenths of an inch. For example:

**;TB + 5, 15D, 30R, 50C, 68**

In this example, the plus sign will be the tab character. Whenever a "+" is found in the text, printing will advance to the next available tab stop. If there aren't any more on the current print line, printing will be advanced to the next print line. It's important to choose the tab character carefully, since it must not occur as normal text. The letter "e", the period, comma, and numbers, usually are not good choices for this. The plus sign "+" and the greater than symbol ">" often are good choices.

The example sets five tab stops. The first one is at "5" (a half-inch in from the left margin). Since no letter follows the number, this tab stop will be "left justified", which means it'll work like a typewriter. This is the simplest, most common kind of tab stop. When a letter does not follow a tab number, "L" (left) is assumed.

The second tab is at "15", and the "D" means the data will be decimal aligned one-and-a-half inches from the left margin. This makes it very easy to line up columns containing dollars and cents. If there's no decimal point in the text characters, the rightmost text character for the tab column will be lined up with the decimal point column. It works even in proportional pitch, as do all the tabs.

The third tab stop is at "30", and the "R" means the data there should be right-justified so that the last text character is indented 3.0 inches. If there's too much text, the right-most characters will be shifted to the next tab column or the next line.

The fourth tab stop is at "50", and the "C" means the text should be centered five inches from the left margin. That's useful for the headings above the columns, but you'll probably have to define a second set of decimal-aligned tabs for the numbers below the headings. Since there's no limit to the number of tab setups you can use in a document, this is straightforward.

The fifth tab stop is at "68", and is a normal, left-justified tab.

If there's too much text to fit between two tab points, the excess text will be shifted to the next tab area, or to the next line. That's just about the same as the operation of a typewriter, except that the excess characters aren't printed one atop the other at the right edge of the platen.

The tabs shown in the example above gave the actual indentations to be used. If you don't give the tabs in ascending order, ALLWRITE will arrange them correctly for you. Tabs can also be specified relative to the preceding tab, by using plus and minus signs. For example:

**;tb + 10, +5, +5, +5**

The actual tab settings would be: 10, 15, 20, 25. One advantage of using relative tabs is that, if you decide to change the starting point ("10" in this example), you won't have to change all the others. Please note that the plus sign "+" can be used as the tab symbol even when relative tabbing is also used, since the two different uses are understood by ALLWRITE.

To turn off all tabbing, just use ";TB" by itself, with no tab symbol or tab stops. One common error is to use the tab symbol as a regular text character, long after the need for tabbing has ended. The solution, of course, is to either re-define the tab symbol or turn it off entirely.

Although most characters can be used as the tab symbol, you should avoid the "at-sign" "@", since it is the default substitution symbol in ";SE" and ";RD".

Here's an example of print-time tabbing. It would be set up like this:

```
;tb + 7,20d,35r
A+Salmon+7.99+sliced
B+Tuna+11.49+diced
C+Halibut+3.79+whole
D+ice+.35+bag
```

It would be printed like this:

A	Salmon	7.99	sliced
B	Tuna	11.49	diced
C	Halibut	3.79	whole
D	ice	.35	bag

One more very special, tab modifier is available. It is the letter "P", and it stands for "pad to here with periods." There can be only one of these in a tab definition. It's intended for use with ALWRITES's "Table of Contents" feature, but can be very useful in setting up certain kinds of catalogs and tables. Here's an example of what we mean:

```
;TB + 10,38P,45D
+widgets++150.20
+anchors++1.59
```

If used with appropriate text, the printed result would be:

Widgets .....	150.20
Anchors .....	1.59
etc.	

Note that all the dots line up, even in proportional pitch.

### **Turning Tabbing Off**

You don't have to turn tabbing off unless you need to use the tab character in normal text, and don't want to assign a different one. In this case, just use ";TB" by itself, followed by an "ENTER" symbol.

### **Pitch-Dependant Tabs**

In 10-pitch and proportional, tab stops are always expressed in tenths of an inch. In 12- and 16-pitch, they can be expressed in column positions instead of tenths of an inch by placing the letter "C" (for "@\$C@%urrent pitch" right after the tab symbol:

**`;TB >C 7,15,31D,46C,60R`**

The "C" in this case is different from the "C" that stands for "centering". ALLWRITE assumes your tab stops are in tenths of an inch unless you use the "C", because it makes it much easier for you to synchronize tabs with hanging indents (a subject we will cover in Chapter 5), and to keep your tabs lined up even when you change pitches from line to line.

If you change pitch within a line, tabs following the pitch change will be calculated in tenths of an inch even if you specified "C".

### **Advantages to Print-Time Tabbing**

The ";TB" control word makes it very easy to revise tab stops: all you have to do is change the tab numbers in one screen line (the one defining the tab stops through ";tb") and then reprint the report. If you use on-screen tabbing, you will have to insert and delete spaces all over the place. Except in the simplest documents, tab stops usually must be changed from table to table, so print-time tabs are much preferred over on-screen ones.

On-screen tabbing only supports left justified positioning, but print-time tabbing also offers right justified, centered, and decimal aligned tabs. Other advantages were mentioned at the start of this topic. The real trade-off is the convenience of being able to line up the columns, right on the screen, versus the flexibility that the screen cannot provide. Print-time tabbing has been used through out most of this book because we're using proportional spaced printing, lots of hanging indents, and setups tailored to the material being presented. Can you imagine trying to do this on a typewriter?

This completes the section on tabbing.

### **MISCELLANEOUS EDITOR CONTROL KEYS**

The few control keys that don't fit neatly into any major category are explained below.

**SCREEN PRINT <\*>**

When dealing with a good word processor, the term "printing" can mean at least two different things: the nice, fully formatted kind of printing that is the end product of your efforts; or a simple listing of some or all of the text, including the control words and emphasis marks. You won't need "as-is" printing too often, but when you're using the Editor just to create a simple list, or are trying to figure out why something isn't formatting correctly, a simple printout becomes very useful.

To print some or all of a file "as-is", you can use the "Block List" method covered earlier in this chapter. If your screen lines are wider than the actual screen, this method will print what is to the right (or left) of the "window" you can see.

To print the text screen just as you see it, press <\*> (Control asterisk). To print the screen when the directory or a help menu is displayed, press the asterisk without pressing CONTROL first. Graphic and control characters will be replaced by spaces, since not all printers can handle them.

If you want to cancel printing once it has started, just press the "BREAK" key: it will stop printing immediately. If your printer has a buffer, it may continue printing after ALLWRITE has stopped sending. Please be aware that, if you aren't using our "ALK" keyboard driver, and your printer is "not-ready", some DOS's will "hang" and not signal the "BREAK" key back to ALLWRITE. If that happens, you must either put the printer into "ready" status or "RESET" the computer. If you have to do a "RESET", you can recover your text afterwards through ALLWRITE's "Warm Restart" facility.

**SPLITTING A LINE INTO TWO LINES -- <ENTER>**

The Editor stores and displays text just as you type it in, except when you use "ENTER" to force a new line, or any of the arrows, control keys, or commands to manipulate the text. Sometimes, after typing something that wound up on a single screen line, you may find the text should be broken into two (or more) lines. The easiest way to do this is to move the cursor so that it is on the first character that should be moved to the next line. If you want to split a line between two words, position the cursor on the blank between them. Then, press the Control key and the "ENTER" key. When you do so, an "ENTER" symbol will appear at the cursor position and the remaining text will be moved to a newly created line just below the cursor line. The "ENTER" symbol is created to keep the two segments apart, since ALLWRITE otherwise would rejoin them, as it joins all short fragments.

**CREATING A BLANK LINE -- <ENTER> and <N>**

There are two ways to create a blank line between two lines of text on the screen. The first of these is to position the cursor at the very start of the second of the two lines of text, and then to press Control "ENTER" (line split). Everything starting at the cursor is split to a new line, and since there's nothing to the left of the cursor, a blank line will be created where the cursor is. An "ENTER" symbol will be created to keep that blank line in place. If you want to type on it, you can do so, or just leave it there so that a blank line can be printed. If you want several blank lines, just keep holding down Control "ENTER", and keyboard repeat will do the rest. When the cursor is at the end of a line, blank lines will appear below the current line. When the cursor is at the start of a line, they will appear above it.

The second way to create a blank line is to position the cursor any place on a line of text, and then press <N> ("N" stands for "new line"). An "ENTER" symbol will be created at the right end of the line, a new blank line, complete with its own "ENTER" symbol, will be created just below the line, and the cursor will be moved to that new line. <N> is not only a nice way of opening some space in the middle of text: it's also useful when defining certain functions in "Soft Keys", a topic that will be covered later on.

**JOINING LINES TOGETHER -- <J>**

As you type, ALLWRITE automatically puts as much on each screen line as possible, and then "word wraps" to the next line of the screen. So, you'll never have half of a word on one line and the other half on the next line; and you won't ever have to be careful about typing past the end of the screen, since the editor will take care of it for you. However, as you insert and delete text later on, some lines will get shorter, while others will get so long that the editor will have to split them to make room for the inserted text. You may even split lines from time to time, using <ENTER>, as explained above.

These editing changes can produce short lines on the screen. If those lines do not end with the "ENTER" symbol, they will be joined together automatically when they scroll off the screen. The next time the text appears on the screen, it'll be shown as full-width lines. This isn't done as soon as you type, because the constant movement of words back and forth on the screen is very hard for the eye to follow, and leads to confusion, if not actual physical distress. (We know, because the editor originally worked that way, and we had to change it because we were getting dizzy watching the screen dance and flicker.)

Now, there may be times when you want the short lines to be joined together immediately. Of course, one way to do that would be to press Shift down arrow once or twice, and then Shift up arrow once or twice. The resulting scrolling would cause automatic joining of text, but it would be awkward if you had to do this. Needless to say, there's a better way.

You can force an immediate "join" of text at any time just by pressing <J>. All text on the screen, except for the lines ending with "ENTER" symbols, will be filled out to whatever is the current screen length.

Using <J> has a second beneficial effect: the text shown on the screen is stored into the main text area in memory, which protects it from the "WHOOPS" command. This way, if you make subsequent changes and "WHOOPS" them out, only changes made after the last <J> and still on the screen will be dropped.

The "ENTER" symbol prevents lines from being joined together, so if you want such a line to be joined to what follows it, just type over or delete the "ENTER" symbol first.

**WORD REVERSAL -- CONTROL ">"**

To reverse two consecutive words, position the cursor any place on the first of the words, then press the Control key and the ">" (greater than) symbol. The ">" was chosen as a reminder that the word will be moved to the right.

**DRAWING A LINE ON THE SCREEN -- Control "-"**

Before showing you how to do this, we must point out that using "emphasis marks" to get underlined text is much better for most purposes. In fact, as distributed, ALLWRITE will place them into the text for you if you press <4> (Control 4) to start underlining, and <5> to end it.

However, you may want a "signature" line in the middle of text, that is of a definite length, in a definite position, and easily noticed on the screen. In those cases, you can draw a blank line on the screen by using <->, (Control minus sign). The "-" is just below the equals sign on the top row of keys. If you hold down the Control key and also hold down the minus, the "underline" symbol, not the minus sign or an "emphasis mark", will appear on the screen. If you keep holding them down, the underline will begin to repeat. Once the line is long enough, just let go to stop.

**HARD SPACES - Control "≡"**

Both the Editor and Text Formatter split lines between words, that is, at spaces. This means that "Mr. Smith" could be split across two lines if "Mr." fit at the end of a line but "Smith" did not. If you don't want a group of words to be split this way, then instead of a regular space, use a hard space.

The hard space character in ALLWRITE can be entered from the keyboard by pressing  $<=>$  (Control equal sign). The symbol that appears on the screen will either be a "plus minus" or a blotchy square, depending on your computer. The actual value of the symbol is "127" and we chose it because most printers cannot print that value (it is the "delete" or "rubout" character). If your printer can print a value for "127" and you need that value, you can assign some other character to stand for "127" by using the ";TR" (translate) control word, as explained towards the end of Chapter 5.

Most other control keys in ALLWRITE are "mnemonics" (first letters of the functions they perform) and therefore very easy to remember. Our apologies on this one, but we've run out of keys! Also, using the equal sign maintains a certain degree of compatibility with NEWSCRIPT, which used SHIFT-CLEAR-EQUAL SIGN to enter the hard space.

**SPECIAL SYMBOLS - Control ":"**

The TRS-80 is able to deal with 256 unique characters (that's that hexadecimal stuff, but we won't mention that word again). However, only about 85 of them are available from the keyboard, and some of the rest are used for arrows, the "ENTER" key, etc. That makes it challenging to get the other 170 symbols into the computer, since there aren't any keys for them.

When you're using ALLWRITE's editor, there's a very easy way to enter all those symbols, including the "ENTER" key, in such a way that they will be accepted as text, not as cursor controls or extraneous symbols. We freely admit that the method was adapted from the IBM Personal Computer: press the Control key, then the colon ":" and then type the three-digit ASCII value of the character you want to use.

The left bracket, "[" for instance, is represented on the TRS-80 keyboard by the up arrow. Since that's a cursor control key, pressing it will move the cursor up, instead of displaying a "[" for you. However, if you consult the ASCII chart in the back of the book that came with your computer, you'll find that the left bracket is an "ASCII 91". So, to get it onto the screen, and into your text for printing, you can press 5 keys in sequence: "Control", "Colon", "0", "9", and "1". Note that the leading zero is required, since a three-digit number between "0" and "255" must be used. Some other common special symbols are listed at the end of this topic.

A couple of comments and cautions are needed now. Five keystrokes sounds like a lot, but special symbols aren't needed too often, and this method is very easy to use: the values can be copied right out of a book, and they are decimal values, but not the other funny kind used by computer wizards.

If you want to use the same special symbol several times in a row, you can just key it in once, and then use "Character Replication", which is the very next topic.

Except for "013", which is the "Carriage return" character, values below "032" (blank) should not be stored as text. Even "013" should be used only in a "SEARCH" or "REPLACE" argument, when you need to manipulate text that includes the "ENTER" symbol, because an "013" placed into the text will always end a line. The Text Formatter rejects all characters below the blank, and will not send them to the printer. If you want to send control characters or graphics to the printer, use the ";SY" (symbol) control word, which is explained towards the end of Chapter 5. We included it for this very purpose.

**Table of Common Symbols**

The list below was printed with the Star SG-10 used to print this manual. Other printers and wheels may print different symbols for some of these values. Since we can't print them all here, we will name the corresponding Epson symbols.

091	[	left bracket	096	*	reverse apostrophe
092	®	reverse slash	123	§	left brace
093	]	right bracket	124		vertical line
094	©	small hat	125	†	right brace
095	_	underline	126	™	tilde

**REPEATING A CHARACTER (REPLICATION) - <K>**

It's easy to repeat any symbol that's on the keyboard, since you can just hold the key down. However, if you've had to use the "literal" control key <::> to create a special symbol, it would be tedious to have to keep pressing five keys over and over again. Instead, all you have to do is position the cursor just after (to the right) of the symbol on the screen, press <K> (Control K), "K" stands for "Key replicate", and the character just to the left of the cursor will be duplicated immediately. If you continue to hold down both the control key and "K", the character will continue to be duplicated. The duplication is done as an "insert" so any text to the right of the cursor will be moved along, not overlayed.

<K> can be used to replicate normal characters as well as special symbols, but it's easier to just hold down a normal key and let it repeat.

**HALF-SCREEN SCROLL UP - <U>**

If you want a line in mid-screen scrolled up to the top of the monitor, put the cursor on that line and press <U> (for "UP"). It's useful when you want to read a paragraph or table that happens to be near the bottom of the monitor at the moment. The cursor doesn't move when you do this, so this control key can also be used for half-screen (or any fractional screen) scrolling.

Believe it or not, we've covered all the control keys now. "Soft keys" work like control keys, but are a separate topic that will be covered at the end of this chapter.

**MISCELLANEOUS EDITOR COMMANDS**

There are a few commands that don't fall into neat groups, and they will be covered next.

**CONTROLLING LINE JOINING -- "JOIN"**

ALLWRITE's Editor "wraps" words from one line to the next to avoid splitting them in the middle, but treats all text as one long series of words until an "ENTER" symbol is encountered. That ends one block, so what follows the "ENTER" is the beginning of another block of text. Within a block, the editor will almost always try to fill out screen lines for you.

Each control word line must be a separate block -- the line before it must end with an "ENTER" (unless the first line of the file contains control words), and the control word line itself must end with an "ENTER".

There may be times when you will not want the Editor to perform this automatic joining of text, but to end each screen line with an "ENTER" symbol instead. The command to use for this is:

**CMD=> JOIN OFF**

The "ENTER" will be added when the text is stored from the screen into memory, not when the cursor moves to the next line. "Join" can be abbreviated to "J", instead of "OFF", you can use "NO". Setting "JOIN OFF" will have no effect on text previously entered, unless you make changes to that text. So, if you set "JOIN OFF", and then scroll through a file, it won't look any different than it did before, and no "ENTER" symbols will be generated. However, if you make a change anywhere on the screen, "ENTER"s will be generated for every line on that particular screen. The reason why the editor works this way is to let you add "ENTER" symbols selectively, rather than throughout the entire file. In actual practice, this command will rarely be used.

To put things back to normal later on:

**CMD=> JOIN ON**

When "Join" is "ON" ("YES" can be used instead), which is the default, "ENTER" symbols are neither created nor deleted automatically: only the ones you type, or the ones contained in certain soft key functions, are stored in the text.

**LINE LENGTH OF ON-SCREEN TEXT -- "LENGTH"**

ALLWRITE lets you control two kinds of line lengths: the one on the screen, and the one that prints. Printed line length is controlled through the ";LL" control word, and will be discussed in the next chapter under the topic, "Margins". Screen line length will be discussed here. It can be set to match the printed line length, the screen width, or some other size.

The editor lets you type lines that are wider (or narrower) than the width of the screen. When you installed ALLWRITE, you were asked to select the maximum and default screen widths. We recommend that the default be the same as that of the actual screen (64 or 80), and that the maximum be 80 unless you expect to work with wide tables. We distinguish between "line length" and "screen width" because they aren't necessarily the same: if you're using a Model III, it has a screen width of 64, but you might have set your maximum line length to "80" or "122".

You can change the current "line length" while editing by using this command:

**CMD=> LEN 70**

"70" is just an example; you can use any number between "20" and the maximum you selected during installation.

As soon as you enter the command, the screen will change to reflect the new line length. If all your text lines end with "ENTER" symbols, nothing will seem to change, of course. If you specify a number (a parameter) that is too small or too large, the following message will be displayed and the command will be ignored:

**MISSING OR INVALID PARAMETER**

"LEN" can be abbreviated to "LE" (but not to "L", which is the abbreviation for "LOCATE"). "LL" (line length) may also be used.

**AUTOSAVE**

This one is easy. As you may recall from the installation procedure, you can tell ALLWRITE to save your text back to disk automatically, after a specified number of changes have been made. The default is "65535", which means "don't do it", but you can set a lower value during installation. However, you can also set an AUTOSAVE value temporarily while running the editor:

**CMD=> AUTO 50**

That would cause ALLWRITE to save your text after every 50 changes. If there were 60 changes when you issued the command, "SAVE" would occur immediately. Any value between 1 and 65535 may be specified. A good choice might be one that caused text to be saved after every ten minutes of typing.

**SELECTING A PRINTER -- "PRINTER"**

This command applies to you only if you have more than one printer, and defined more than one printer when you installed ALLWRITE. If that is the case, the printers are numbered "1", "2" and "3"; and the name of the current printer appears on the right hand side of the status screen. When ALLWRITE starts, printer number 1 is always selected. If you want to use one of your other printers (and it's connected to the computer), you can just use this command:

**CMD=> PRINTER 2**

"P" is an acceptable abbreviation, and the number could be "1", "2", or "3". If you didn't define a printer for the number you enter, the status screen will show "none" for the printer, and attempts to print anyway are likely to result in errors. If you give a number below "1" or above "3", an error message will be displayed, and the command will be ignored.

Once selected, the printer will remain current until ALLWRITE is ended or another printer is selected.

**REPEATING AND DISPLAYING THE PREVIOUS COMMAND**

If you want the Editor to perform the last command again, the command to use is the "equal sign":

**CMD=> =**

If you want the Editor to display the last command, and allow you to modify it before repeating it, the command to use is the "question mark":

**CMD=> ?**

As soon as you've typed the "?" and "ENTER", the previous command will appear on the command line, with the cursor at the right side. You can move the cursor back and forth on the command line to modify it, then press "ENTER" to execute the modified command. If the command was O.K. the way it was, just press "ENTER" to execute it again.

If you don't want the previous command repeated after seeing it, press "BREAK" to cancel the command mode.

**VALIDATING EMPHASIS MARKS BALANCING -- VA Command**

The Editor can check to see whether certain formatting features are "balanced", that is, turned off after they've been turned on. The features it checks are:

**Underlining      boldface      double-width      italics**

The command to use is:

**CMD=> VA [symbol]**

where "symbol" is the emphasis mark you are using. If you don't specify a symbol, the at-sign, "@", will be assumed by default.

Validation takes less than a tenth of a second, so you won't detect any delay between the time you press "ENTER" and the time the cursor re-appears when the command is done. If every feature that is turned "ON" is turned back "OFF" before the next "ON", and if they are all "OFF" by the end of the file currently in memory, nothing will seem to happen: that means everything is O.K. However, if an imbalance is detected, the text containing the first unbalanced "ON" emphasis mark will be brought onto the screen, and a large rectangular cursor will mark it for you. If you want to correct the error, press "BREAK" and the cursor will revert to its normal shape. If you want to leave the imbalance and keep looking for other imbalances, press the "ENTER" key.

Since each Emphasis Mark is independent of any others, the validation scan may seem to back up when you use it; but that's only because of the sequence in which things are checked.

When the cursor returns to its normal shape (a square), it means validation is done. If you make any corrections, you can use the command again to make sure there aren't any other errors. Note that by pressing "ENTER", you can get past intentional errors to look for other ones. If more than one condition is "ON" at the end of the file, only one of the conditions will be marked; after you correct it and retry "VA", the next one will be shown.

**Limitations**

The "Validate" command doesn't recognize ";ES" changes, emphasis marks that start in one file and end in another one, or any conditions other than the ones listed above. It can be used with DOTWRITER files as well as with ALLWRITE files. Because it's so fast, we recommend you make a habit of using it, especially with large files or DOTWRITER files. You can get a complete text validation by using the ";VI" control word as a run-time option of the Text Formatter. Note that ALLWRITE's "italics" is DOTWRITER's "alternate font".

**COMMANDS TO CONTROL FILES**

The Editor has about a dozen commands that let you move from one file to another, display the directory, or save text. Some of them were covered in Chapter 3, and the rest will be explained below.

The file-oriented commands in Chapter 3 were:

**DIR      SAVE      QUIT**

**SAVE**

The "SAVE" command stores text from memory to disk and then re-displays the text so you can continue editing it. The command has two variations that were not explained in Chapter 3:

1. The file can be saved under a different name:

**CMD=> save copy2/doc:2**

2. The file can be saved on a different drive:

**CMD=> save :3**

The permanent filename stored in ALLWRITE is not changed by this command. If you issue another "SAVE" later on, with no name or drive specified, the permanent name will be used.

When only the drive number is given, the permanent file name is used, but the drive number is changed temporarily to the one you've specified. It doesn't matter whether or not the permanent name included a drive number.

As shown in Chapter 3, if "SAVE" is used without any parameters at all, the file is saved under its permanent name, thereby replacing the previous copy on disk.

**KILLING (REMOVING) A DISK FILE**

The "KILL" command can be issued directly from ALLWRITE, and does the same thing as the DOS "KILL" and "REMOVE" commands:

**CMD=> KILL LETTER/OLD**

**CHANGING THE PERMANENT FILE NAME**

Each time you start running ALLWRITE, it will ask you for the name of the document to be edited. Your answer is stored as the permanent name of this file, and will be used whenever the text is saved to disk, unless temporarily overridden as shown above, under "SAVE".

If you want to change the stored permanent name so that subsequent "SAVE" commands, even without any parameters, will use that different name, the command to use is "NAME":

**CMD=> NAME extra/cpy:3**

That will change the permanent name to "extra/cpy:3", and all future "SAVE" operations will use that new name by default. The original file on disk is not affected by this.

The command can be abbreviated to "N". If you don't give a file name, the STATUS screen will be displayed to show you the current permanent name.

### EXITING FROM ALLWRITE

There are five ways to exit from ALLWRITE:

1. the Editor's "QUIT" command, explained in Chapter 3;
2. the Editor's "END" command, explained below;
3. the Editor's "RUN" command, explained later on;
4. the Text Formatter's exit menu, when printing ends;
5. resetting or turning off the computer.

We recommend against using the "reset" method, since disk files may be open at the time.

#### The "END" Command

When you've finished writing or editing a file, and want to save it and then edit another file or exit to DOS (without printing first), you can do it in one step:

**CMD=> END**

The file is saved under its permanent name by default, but you can specify an alternate file name or drive number to be used instead.

**CMD=> en newcopy**  
**CMD=> en :2** (the colon is required)

"EN" is the shortest acceptable abbreviation for "END". After the file has been saved, ALLWRITE will prompt you for the name of the next file to be edited. If you want to edit something else, type its name. If you want to exit to DOS, just press "BREAK".

### SWITCHING FROM ONE FILE TO ANOTHER

When you've finished editing a file, you can save and print it (Soft key capital 1 or 2), save or quit and exit to DOS, or start editing another file. We've explained how to save, print, and exit, and now we'll show you some ways to switch editing from one file to another. One very important method, called "Linking", will be discussed a little bit later on; it's an automatic way of switching files, whereas the ones shown below are controlled by you.

#### The "NEW" Command

If you want to switch to another file, but don't want to save the current one first, use "NEW":

**CMD=> NEW**

If the current file contains unsaved changes, you'll be asked whether it should be saved before another file is edited. Otherwise, you'll just be asked for the name of the next file. This command is similar to "END", but doesn't automatically write the current file to disk.

**RUNNING ANOTHER PROGRAM -- The "RUN" Command**

When the Editor ends, it can start almost any other DOS program, such as "BACKUP" or "TALLYMASTER". The command to use is "RUN", and the extension portion of the next program's name (usually "/CMD") must be included in the command you issue:

**CMD=> RUN BACKUP/CMD**

If there are unsaved changes, you'll be asked whether the current file should be saved. If the program you specify isn't found, the Editor will display an error message and stay with the current file. If you omit the "/CMD", the program you specify probably will not be found.

The RUN command does not pass command line parameters to the next program, so this won't work:

**BAD EXAMPLE:**      **CMD=> run backup/cmd :1 :2**

"BACKUP" will run, but it will still ask you for the drive numbers, since ":1 :2" will not have been passed to it.

The "RUN" command's main purpose is to integrate the Editor with other software packages such as Electric Webster and Dotwriter, and to transfer control to ALLWRITE's Text Formatter (the printing program). However, it's flexible enough to let you run other things as well.

**The "OPTIONS" Command**

This is a special purpose command, and you probably won't ever use it unless you decide to re-define certain "Soft Keys". In Chapter 3, we explained that Soft Key capital 1 runs the Text Formatter automatically, while Soft Key capital 2 runs the Text Formatter in a way that lets you select print-time options. To do this, the editor passes a signal to the Text Formatter. The signal is set by the "OPTIONS" command, which recognizes either of two letters as parameters:

**CMD=> OPTIONS R**  
**CMD=> OPTIONS P**

"R" stands for "Run automatically" and "P" stands for "Prompt". If you ever want to use this command, do so before issuing the "RUN" command discussed earlier, since "RUN" passes the "OPTIONS" command to the Text Formatter.

**EDITING 3 FILES AT A TIME (Model 4 only)**

This feature only works on 128K TRS-80 Model 4s, 4Ps, and any "workalike" machines that function identically to these. It does not work on other machines, even if they have more than 48K of memory.

If you have a 128K Model 4, the extra 64K of memory is divided into 2 banks of 32K each (this is explained in the book that came with your computer). You can use each of those banks as a "Spooler" (internal print buffer), a "MEMDISK" (internal, small, high-speed disk drive), or, with ALLWRITE, to edit more than one file at a time.

Normally, when you're editing a file with ALLWRITE, the text is stored in the primary bank, and some of it is stored in the System bank along with DOS and ALLWRITE itself. The total text space available to you is about 34,000 characters (if you're using high memory drivers from DOS, or ALK with DOSPLUS, you may have less than this available). If one or both of the other two banks is available, and you want to edit a 2nd or 3rd file in that extra space, this command can be used:

**CMD=> AREA 2**

The number following the command can be "1", "2", or "3", and if no number is given, "1" (the original, primary area) is assumed by default. The command can be abbreviated to "AR". We chose the word "area" instead of "bank" because the 1st area is bigger than a single bank.

If the 2nd or 3rd area is available when you ask for it, you'll be prompted for the name of the file to be edited in that area:

**ENTER NAME OF DOCUMENT TO BE EDITED:**

That's the same prompt you've seen when starting ALLWRITE. You can ask for a Disk Directory by typing a "?" mark, reply with a name, or press the minus sign to cancel the request and stay with the primary file you were already editing.

If you give a file name and the file is found, it'll be read into the specified area and then displayed. The second and third areas are a little smaller than the 1st: a bit under 32,700 characters apiece.

You can determine which area you're in at any time by pressing the STATUS key: the area number is displayed right at the top.

The AREA command can be used again to bring a file into the 3rd area. If the file you specify doesn't exist, you can create it as a new file, just as you can in the primary area you've been using until now.

To switch from one area to another, just use the AREA command, followed by the number of the area in which you want to work. Once a file has been assigned to an area, switching from one to another occurs instantaneously. This makes it very easy to see what's in each area.

If you plan to use Areas heavily, you might find it worthwhile to define 3 soft keys to stand for this command; it'll make switching back and forth very quick and easy.

The text in each area is maintained independently of the others, so changes in one area, even global changes, don't affect text in the other areas. However, any "search arguments" defined in one area will still be available when you work in a different area.

This is also true of named points: you should avoid using the same points in different areas, or you may get unexpected results.

If a system failure occurs while Area 2 or Area 3 is active, it may not be possible to salvage all text from memory with a "Warm Start". If Area 1 was on the screen at the time, all text should be salvageable if it wasn't destroyed, but if either of the other areas was on the screen, text in the 1st area may not be recoverable.

**Moving Text Between Areas**

This can't be done directly, because the "MOVE" and "COPY" block keys do not work between 2 areas. Instead, you can "PUTFILE" from an area to disk, and then "GETFILE" from disk to a different area. Our decision not to support direct inter-area block moves was based on the large amount of extra memory that such a feature would have consumed, the low frequency with which we estimated people would use such a feature if it existed, and the potential danger of losing text that was placed in a temporary "cut and paste" area.

**Exiting with Multiple Areas**

When you issue any of the "exit" commands ("END", "QUIT", "RUN", or "LINK"), all 3 areas will be checked for unsaved changes. If any are found, the STATUS screen for the next changed area will be displayed and you will be switched automatically to that area. This is a safety precaution: if the editor just ignored the changes in other areas, they would be lost.

If you want to save the changes, just use the "END" command when this happens; if you don't want to save them, use "QUIT" and confirm it when asked. If there are changes in more than one area, this automatic switching will happen more than once.

### Passing a Text File to the Formatter

Only the name of the file edited in the primary area (Area 1) will be passed to the Text Formatter, Dotwriter, or Electric Webster. So, even if you've been editing in Area 2 and press Soft Key capital 1, the Formatter will print the file that had been edited in Area 1. The solution to this is very simple: invoke the Formatter with Soft Key capital 2, which tells the Formatter to let you select options, including a different file.

We've covered all the file manipulation commands but one, and that command is so important that it will be covered as a separate topic, coming up next.

## LONG DOCUMENTS

To be as fast as possible, ALLWRITE's editor works only with text kept in memory (RAM), rather than reading and writing constantly to disk. This method works very well with documents small enough to fit in memory, but would seem to place a severe limitation on attempts to write anything longer than 10-15 single spaced pages. Now, this book is a lot longer than that, and since it was written and printed using ALLWRITE, there must be a convenient way to get around the limited memory size of the computer. And that is what we will explain to you here.

This section will cover several related topics. Some of them are likely to be used by most people, and some will only be needed in special cases. When 1st reading this section, we suggest you just learn about "APPEND", "BACK", and "LINK", all in their simple forms. Their fancier uses, covered afterwards, probably can be skimmed, just so you know the capabilities are there if you ever need them in the future.

There are 2 ways to set up long documents. These ways can be used separately or together, depending on your needs and preferences. The 1st way is to "link" separate disk files together to form a document of unlimited length. These links can be made both forward and backward, forming a chain of files. We will call this an "append" chain, because the control word used to connect them in the forward direction is ";AP", which stands for "append". The 2nd way is to let the main file reference another file (or part of another file) that is to be "imbedded" at print time within the main file. If you think of the "append chain" as a real chain of metal links, then the linked (appended) files form the chain itself, while the imbedded files dangle from the main chain, without going further.

Both the Editor and the Formatter recognize the "link" method, but only the Formatter recognizes the "imbed" method. Since the "imbed" method is generally used to incorporate pre-written "boilerplate" or standard layout specifications, this 2 fold scheme gives a great deal of flexibility. First, we will show you how to "link" files together.

### THE "APPEND" AND "BACK" CONTROL WORDS

When the size of the file you're writing approaches the size of memory, (25,000 - 26,000 characters on a Model I or III; 33,000 - 34,000 characters on a Model 4), and you reach some kind of logical break in subject matter, just press "ENTER" to close the final paragraph, and then end the document with the control word shown below, followed by the name of the file that will contain what you will write next:

**;AP part2/doc**

"AP" stands for "append", and tells the Formatter to switch to the referenced file and then continue as though the text in it was part of the current file. Any valid file name can be used, but you must avoid using a file that appeared earlier in the series of links you've been creating. The chain you build this way can be as long as necessary, and the files can be on different disks. There's no upper limit to this: if you had enough floppy diskettes, you could process an encyclopedia.

If the next file isn't found, you'll be prompted to enter the correct "name" in the linked chain of files. This will enable you to switch disks, if necessary, or to type the name correctly, if it was mis-spelled in the file. If there is no next file so far, you can reply with a minus sign "-" to cancel further printing.

The Editor recognizes ";AP" only if it is the very last thing in the file. An "append" in that position enables the Editor to switch from the current file to the appended one. We'll explain how that's done after we show you how to set up a backwards link to the file that precedes the current one. The control word is ";BA":

**;BA part1/doc**

If used, this control word must be the very first thing in the file. It must appear before comments, setups, titles, or anything else. It is used only by the Editor, since the Formatter only can print a document going forward (one of its few limitations).

If either ";BA" or ";AP" appears correctly in the current file, the file names they reference will appear on the "STATUS" screen above and below the name of the file you're editing now.

";AP" and ";BA", along with the file names that follow them, must be the only things on a control word line. The first file, which is the starting point of the chain, won't have a backward link, of course, since nothing precedes it; and the last file of the chain won't have a forward link, because nothing follows it.

**Recommended File Sizes**

Files do not have to fill up memory before appended files are started. It takes longer to read and write a large file than a small one; and if you've left some room in a file, it will be much easier to add text to it later on. We recommend leaving at least a couple of thousand characters of space in each file for future growth. However, it's easy to split a file if it becomes too large: that's one of the uses of the "PUTFILE" control key.

If you run out of memory while editing, see "Error 251" in Chapter 7. It explains how to split the file in two.

**THE "LINK" COMMAND**

Now that we've explained how to set up the links between adjacent segments of a document, we can show you how to use them. The command is:

**CMD=> LINK F  
CMD=> LINK B**

"LI" is the minimum abbreviation for "link", "F" stands for "Forward", and "B" stands for "Backward". "F" is the default if you just type "LI". Link Forward will look for an ";AP fileid" at the end of the current file. If it finds it, it will attempt to switch from the current file to the next one, and if there are unsaved changes in the current file, you'll be shown the standard message asking whether or not to save those changes.

If there is no ";AP", or if the referenced file isn't found, an error message will be displayed and the Editor will re-display the current file. No harm will be done.

Link Backward looks for a ";BA fileid" at the start of the current file, and works just like Link Forward, except that it goes in the opposite direction along the chain of files.

### Using The LINK Command

In actual practice, the forward and backward links are used by simply pressing the Soft Keys we assigned to them when we made up the ALLWRITE distribution disks. If you want to use those soft keys for something else, you can always perform a link by pressing BREAK and typing it as a command.

To link backward, press: SOFT KEY CAPITAL 5 <%>

Once you've typed the necessary links into your files, letting the Editor move from one segment to another is as easy as pressing a couple of keys.

Now for the best parts: this method is very fast and easy to use. Linking enables you to save many changes, and then have the Editor automatically read in the next or previous segment. If there aren't any changes, the save step is skipped. Since saving and verifying a large file to floppy disk takes about 20-25 seconds, and reading one takes about 5-6 seconds, this means you can move through a large document at the rate of 6 or 30 seconds per 4,000 words (depending on whether or not changes have to be saved back to disk), and even those short delays are entirely under your control. You won't find the Editor stopping unexpectedly to write a little text out to disk and then read in a little text. The links happen only when you want them to happen.

Another nice thing about the method is something we've mentioned before: all your search and replace arguments, screen settings, tabs and soft key definitions remain intact for re-use, just as though you were still editing the same file and had scrolled forward or backward in it. Of course, that's exactly what's happened, and the editor just helped you instead of getting in your way.

This method is inherently open-ended: your documents aren't limited in size to what fits on one disk. For example, this book fills four double density 40-track diskettes, but it was easy to write, preview, and print draft copies or portions of it. The final copy, including the Table of Contents and Index, was generated in just one pass. The only thing we had to do were change disks every couple of hours, and change paper and ribbons when they ran out. Nice.

If you're writing a book or a long report, you can print portions of it while it's being developed. You don't have to start from the beginning each time. While writing this book, we "previewed" the material on the screen to catch and correct simple formatting errors (such as forgetting to turn underlining off). Then, we printed the latest 30 pages or so, as the writing progressed. From time to time, we printed a full copy so that several people could proof-read what had been written so far, then used the Editor to make the changes in the computer files. Once a section seemed O.K., it didn't have to be printed again until the final stages were reached.

The final big advantage of this method is safety: the files are normal DOS files, and each one can be backed up and copied independently of the others. That means the loss of a little data won't result in the loss of an entire document, let alone an entire disk.

### CHOOSING FILE NAMES

Although ALLWRITE will follow the links regardless of the names you choose for the files involved, it'll be easier for you to keep track of what you are doing if the names pertain to the material they contain. For example, a few of the files used to write this book were:

**TUTCMD/CH3**  
**TUTFMT/CH3**  
**TUTEDIT/CH4**  
**TUTSRCH/CH4**

In "TUTFMT/CH3" the links were:

```
;ba tutcmd/ch3
...
...
;ap tutedit/ch4
```

### IMBEDDING FILES -- IM Control Word

The Editor's "GETFILE" command is used to incorporate text from one file into another. That's useful when you must change some text but want to keep the original file for other uses. However, if you don't need to make any changes, making multiple copies of text by "GETting" it into several other files just wastes disk space. It can also lead to confusion if you accidentally start making changes to some versions but not to others.

A better way to use "boilerplate" material or a standard layout is the ";IM" ("imbed") control word:

**;IM stock/par**

Any valid file name can be used, of course. The Editor doesn't recognize ";IM" the way it does ";AP", so it can appear anywhere in your documents, as long as it's the last entry on a control word line.

When the Formatter encounters "IMBED", it notes where it is in the current file, then switches temporarily to the imbedded file. It reads and prints the contents of that imbedded file just as though it had been the next portion of the current file. When the end of the imbedded file is reached, the Formatter switches back to the current file, and continues from where it left off in that current file.

";IM" does not cause a "control break," so if there are no control words at the beginning of the imbedded file, it's text will immediately follow the text just before the ";IM" in the main file.

### Nested Imbeds (Imbeds Within Imbeds)

An imbedded file can contain an ";IM fileid", so you can develop sophisticated (a nice way of saying "complicated") chains of files if necessary. The limit on nested imbeds is four, not counting the main file. Anything past that is an error.

### Example of "IMBED"

In the example below, the main file is on the left and the two files it imbeds are on the right:

The super-broom is like	;cm FEATUR1
nothing you've ever seen	* Ergonomic handle
before	;br
;sk	* genuine straw
;im featur1	
;im featur2	
;sk	;cm FEATUR2
And it's available now!	* available in 7 colors
	;br
	* guaranteed 90 minutes

When printed by the Text Formatter, this incredible landmark in advertising literature will look like this:

The super-broom is like nothing you've ever seen before"

- Ergonometic handle
- genuine straw
- available in 7 colors
- guaranteed 90 minutes
- And it's available now!

### **USING SEVERAL DISKETTES -- ST Control word**

If a document is very large, it may not fit on one diskette. When that happens, the next series of files in the chain can be kept on a 2nd diskette, or on as many additional diskettes as are needed. There's no upper limit to this, assuming you have enough paper and ribbons to keep things running when the time comes to print your encyclopedia. However, when the ";APPEND" at the end of the last file on the first disk is encountered, and the next disk isn't in a disk drive, ALLWRITE won't find the next file in the chain (mostly because it isn't in a disk drive yet).

There are two ways to handle this. If you just let it happen, ALLWRITE will tell you that the file wasn't found, and ask you to type in its name. You can switch disks at that point, press "ENTER" (without even re-typing the name), and let ALLWRITE look for the file again.

The other method is a bit more organized: You can place the ;ST (;stop) control word on the line before the ";APPEND":

***;ST any message you like***

"ST" causes ALLWRITE to stop reading from disk, display your message, and then wait for you to press the "ENTER" key before resuming. Printing will continue while the disks are being switched, since there will still be text in memory. You can switch disks and then press "ENTER", because in this situation, ALLWRITE will look ahead, past the ";ST", and store the ";AP" just in case you do switch disks. Then, the appended file will be started, just as though that next file had been available in the first place. For example:

***... and waited for Dawn.  
;st \*\*\* SWITCH TO DISK #3, PRESS ENTER \*\*\*  
;ap chap19***

For this to work properly, ";AP" must be on the very next screen line following ";ST".

### **NOTE OF CAUTION REGARDING "APPEND"**

If a chain of "appends" references itself, the Text Formatter will wind up going in a circle. Although we could have checked for the simplest kind of circular reference (a file that appends itself), it didn't seem likely that you would make that mistake (unless you did it on purpose, to see what would happen). But, there's really no way for ALLWRITE to make sure that the thirty-seventh file in a chain of "Appends" doesn't append the ninth file of the chain. So, if you develop anything large, this kind of error is one to avoid. It won't hurt your disk files, but will use up a lot of paper, especially if you start printing just before you go to bed.

**IMBEDDING ONLY PART OF A FILE**

This section will explain a more advanced use of "APPEND" and "IMBED", and after reading this paragraph, you may want to skip the rest of this section for the time being. One of ALLWRITE's control words is ";LB", which stands for "LABEL", and it lets you put unique identifiers in a file. APPEND and IMBED can reference those labels so that only a portion of the referenced file will be imbedded or appended. This capability is useful when extracting boilerplate or stock setups from a file that acts as a master library. We've supplied a file of this sort with ALLWRITE; it's called "ALF/DEF" (for "ALLWRITE Formatter defaults") and it provides the default layouts for Table of Contents, index, Legal Documents, and Mailing Labels. All those setups are small, and by storing them in a single file, we were able to save a fair amount of disk space.

**Labeling Blocks of Text -- LB and EN Control Words**

Still with us? Well, it's actually fairly easy to use these selective capabilities, because they only make use of three new control words. Here are two of them:

**;LB label**  
**;EN label**

";LB" stands for "LABEL", and it marks the beginning of a particular block of text. "label" is anything you want to use as a label. It must begin with a letter ("A" through "Z") and not be more than eight characters in length. The other characters may be letters or numbers. Upper and lower case are treated alike ("A" is the same as "a"), and the first character that isn't a letter or a number ends the label.

";EN" stands for "END", and marks the end of a block of text. The "label" must match the one used with "LB". If it doesn't, ALLWRITE will continue imbedding text until it finds a match or reaches the end of the file.

**CAUTION:** If ";EN" is used without a "label", it will mark the end of text from the file in which it occurs, and the Text Formatter will not read past it. Period. There's another facility in ALLWRITE that makes use of that, but it isn't relevant to the current topic. For safety, if an ";EN" is supposed to match an ";LB", put the same label on both of them.

";LB", followed by a label, must be the first and only control word on the line. ";EN" must be the last control word on the line, and for clarity in reading what you have done, should be the only control word on the line.

You can find several examples of these control words in the file "ALF/DEF".

**REFERENCING LABELS -- IM, AP, and GO Control Words**

The reason for using labels within a file is to enable ALLWRITE to pick a starting point other than the top of the file, and an ending point other than the end of the file. There are 3 ways to make use of those labels:

<ul style="list-style-type: none"> <li>• <b>in an IMBED:</b></li> <li>• <b>in an APPEND:</b></li> <li>• <b>in a GOTO:</b></li> </ul>	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;"><b>;IM stock, greeting</b></td> <td style="width: 50%;"><b>;AP terms, partner</b></td> </tr> <tr> <td style="width: 50%;"><b>;GO topic7</b></td> <td style="width: 50%;"></td> </tr> </table>	<b>;IM stock, greeting</b>	<b>;AP terms, partner</b>	<b>;GO topic7</b>	
<b>;IM stock, greeting</b>	<b>;AP terms, partner</b>				
<b>;GO topic7</b>					

"stock" and "terms" are the names of files to be imbedded or appended. "greeting" and "partner" are the starting points to be used in the respective files. All text before each label will be ignored, and the labels themselves won't be printed. If a label isn't found, the entire file will be read but skipped.

";GO label" does the same thing as a selective ";AP label", but within the same file. The "label" must appear with an ";LB" someplace after the ";GO". If you refer to a label that appears earlier than the ";GO", it won't be found. If a label isn't found, the rest of the file will be skipped. If you want to reference a label that precedes the ";GO", use ";AP" instead, and append the current file to itself. Of course, that's a circular reference, so it can lead to problems if not used properly.

The purpose of ";GO" is to provide an easy way of skipping the printing of certain passages of text for the time being, without having to delete that text from the file. Later on, the ";GO" can be changed to a comment ";CM" or deleted, and the text it skipped will be printed thereafter. Note that skipped text is entirely ignored by the Text Formatter during printing, and any control words within that text will have no effect on the formatting setup of your document.

This completes the section on long documents, linked files, and selective appending and imbedding.

## **SOFT KEYS**

A "Soft Key" is a single key that can be defined to stand for several keystrokes, and then, when pressed, pass them through the Editor for processing. This can save time and improve accuracy, since only one key, instead of several, needs to be pressed. ALLWRITE's Editor has 22 soft keys. This section will explain how you can define their contents, store those definitions on disk, and use them afterwards. We'll also show you how you can create what amounts to a command file that can customize the Editor each time you start to run it.

We pre-defined most of the soft keys for you, but you can change them whenever you want to do so. If you just want to use them as-is, you don't have to read the rest of this section: just look at the Soft Key Cue card, or the on-line HELP topic, "SOFT DEFAULTS." If you do want to change them, you can save your new settings in "AL/DEF", which is read automatically by the Editor each time it starts to run. If you want to define and save several sets of definitions, you can use several disk files, each with a different name.

### **Using A Soft Key**

We'll mention this first, since it may be all you need to know for the time being. Soft Keys are used like normal control key functions such as <D>: press the CONTROL key and then the Soft Key you want to use. Its definition will be processed by the Editor just as though you had pressed each of the stored keys individually. If the Soft Key isn't defined, it will be ignored, and no harm will be done. Some Soft Keys issue commands for you, so the bottom line of the screen may flicker momentarily as the command prompt appears and vanishes.

### **What Can Be Stored in a Soft Key**

Each soft key can store up to 22 keystrokes -- any keystrokes, including text, arrows, control keys, the BREAK key, and references to other soft keys. Since even the arrows can be stored, this means you can't correct a definition once you begin it, except by ending the incorrect definition and redefining the entire soft key over again. Defining one soft key has no effect on any of the others, but once you start defining a soft key, its previous definition will be lost immediately.

In the interest of safety, a few things can't be stored in a Soft Key. Some Editor commands and control keys prompt you for further information, or for confirmation that it's O.K. to continue. It wouldn't be safe to let a Soft Key supply the answer in these cases, so the Soft Key definition is suspended while the Editor waits for you to reply from the keyboard. Once you do so, the rest of the Soft Key definition continues to be used.

The Soft Keys only work while you are editing; they don't work when running the Text Formatter, or when you've ended ALLWRITE and gone back to DOS.

**How To Define a Soft Key**

The command used to define a Soft Key is:

**CMD=> KEY x**

"KEY" can be abbreviated to "KE", and "x" stands for the key you actually want to define. It can be any of these 22 keys:

**1 2 3 4 5 6 7 8 9 0 ! " # \$ % & ' ( ) X Y Z**

The three letters "X", "Y", and "Z" can be in upper or lower case; "X" and "x" are the same thing. These three Soft Keys can be defined and used just like the others, but have an extra capability that will be explained later on: they can be used as commands with a repetition value.

When you enter the "KEY" command this way, the screen will clear and you'll see this prompt:

**DEFINE KEY -X-**

"X" is just an example; in actual use, the key you pressed will be shown. The cursor will be positioned to the right of the prompt, and a graphic rectangle will appear somewhat farther to the right on the screen. It shows you where the definition must end: you can stop before reaching it, but you cannot type into or past it.

Once the prompt appears, you can type whatever sequence of keystrokes you wish. If your definition is shorter than 22 characters, it must be ended by the special indicator, CONTROL "Q" (don't use the quotes): <Q> is how we've been showing that sort of thing throughout this book. If you type right up to the graphic boundary, the definition will end and be stored automatically. A small problem will arise if your definition is 21 characters in length, since it takes two keystrokes (CONTROL and Q) to end a short definition, and only one keystroke will be left. The possible solutions are to figure out how to make the definition one keystroke shorter, or one keystroke longer, or to continue it in a second soft key as explained later on.

As you press each key that is to be included in the definition, it will appear on the screen. The arrows, BREAK, and CONTROL <sup>1</sup> keys will appear as special symbols that are computer dependant. For example, the "ENTER" key will appear as a capital "M" on the Model 1 (ENTER is the 13th ASCII character, and "M" is the 13th letter), but as a funny looking "R" on the Models III and 4. The screen representation won't affect their proper use, however.

Here are 4 examples of Soft Key definitions. In each case, the "KEY" command and the key to be defined must be pressed, followed by the "ENTER" key and the definition shown. Since we can't print the symbols for arrows, BREAK, etc., we will use the names of the keys in question, and explain the definitions immediately afterwards.

1. If the "CLEAR" key doesn't work, use SHIFT CLEAR or F1.@=

Example 1: A very long, frequently used word can be stored in a soft key to reduce typing time and improve accuracy. If you had to refer frequently to "photochromograph", (a high speed movie camera) you could store it in Soft Key "8" this way:

**photochromograph CONTROL q**

That would be the 16 letters of the word, then the Control key, then the letter "q", where <Q> marks the end of the definition. Eighteen keystrokes were used, and that is less than the 22 keystroke limit.

Example 2: Soft Key 1, "paragraph", is defined this way:

**CONTROL n;pp CONTROL n CONTROL q**

<N> guarantees an "ENTER" symbol at the end of the previous line, and creates a new blank line. ";pp" is the control word used by ALLWRITE to start a new paragraph. The second <N> places an "ENTER" after the ";pp" and starts another blank line where the user can type more text. <Q> ends the definition, so nine keystrokes were used.

Example 3: Soft key 4, "underline on", is defined as:

**CONTROL o @ CONTROL o \$ CONTROL q**

<O> forces a one-key insert, so the next keystroke will be inserted regardless of whether the Editor is in overlay or insert mode when the soft key is used. The "@" sign is the inserted keystroke, and begins an Emphasis Mark. The second <O> forces another 1 key insert, and the "\$" completes the Emphasis Mark for underlining. Finally, the <Q> ends the definition, so eight keystrokes were used. The one-key insert method was chosen for two reasons: we wanted to insert, not overlay, the Emphasis Mark; and we couldn't be sure whether insert mode would be on or off at the time. If we had tried to use <I>, and it turned out that insert mode had already been in effect, <I> would have toggled the Editor back to overlay mode.

Example 4: Soft Key Capital 1 (<!>), used to invoke the Text Formatter for automatic printing, is defined as:

**BREAK op r ENTER BREAK run alf/cmd ENTER CONTROL q**

"BREAK" places the Editor into command mode. "op r" means "options run", and is a command that sets the "automatic" signal for the Text Formatter. "ENTER" ends that command and causes it to be accepted and executed. The second "BREAK" starts another command. "run alf/cmd" is that command, and it tells the Editor to run "alf/cmd", which is the Text Formatter. The second "ENTER" ends and executes that command, and <Q> ends the Soft Key definition. 21 keystrokes were used. Note that we couldn't spell out "options", since it would have required more than 22 keystrokes.

### Long Definitions

You can set up a definition longer than 22 keystrokes by chaining several soft keys together. For example, the definition could begin in <X>, which would contain the first 21 keystrokes, followed by two other keystrokes <Y>. The rest of the definition could be stored in <Y>, which would end with <Q>. It will take two "KEY" commands to do this. To use the result, just press <X> when it's needed. The contents of <X> will be used, and it will chain to <Y>.

### SAVING SOFT KEYS TO DISK -- The "KEY" Command

All 22 current soft key definitions can be saved into a disk file and loaded for use at any later time. The command is:

**CMD=> KEY SAVE fileid**

"SAVE" can be abbreviated to "S", and "fileid" is any valid file name acceptable to your DOS. Of course, you shouldn't use a name already assigned to some other file, unless you intentionally want to replace that other file. When the command is used, all the current soft keys are saved; you can't save just a couple of them. If you want to replace the standard definitions we've supplied, use this command:

**CMD=> key s al/def**

This file is read in automatically every time the Editor starts, so the definitions it contains are the ones that will be available automatically. If you store definitions in a different file, you can load them back in for use by using this command later on:

**CMD=> KEY LOAD fileid**

"LOAD" can be abbreviated to "L", and "fileid" should name a file into which you have already stored some definitions. If you "LOAD" some other kind of file, results will be unpredictable, since ALLWRITE can't really check the contents to make sure they are valid.

**Repeating Soft Keys**

One way to use the contents of a soft key several times is to press the key several times. However, if you want to repeat a soft key a varying number of times, the "X" and "Y" commands can be used for the purpose. "X" and "Y" must be defined as normal soft keys first, and they can be used as normal soft keys afterwards. When you want one of them to be executed several times, you can use it as a command:

**CMD=> x 7**

That would cause <X> to be processed 7 times, just as though you had pressed it 7 times. If <X> stood for "Hi", the result of the above command would be:

**HiHiHiHiHiHiHi**

<Z> is a little different from <X> or <Y>. When used as a control key, it's just like any other soft key, and must be defined in advance. When used as a command, it executes <X> and then <Y>, one after the other. This means you can execute two commands in alternating sequence. And, since you can supply a repetition factor, you can repeat the alternating sequence several times.

For example, if these were the definitions of <X> and <Y>:

**X: 123 <Q> Y: 456 <Q>**

and you issued this command:

**CMD=> Z 4**

this is what would appear on the screen:

**123456123456123456123456**

That contains "123" and then "456", four times.

**Limits to Repetition**

The "X" and "Y" commands can be repeated at most 32 times; the "Z" command can be repeated at most 16 times, since it invokes two other commands, for a total of 32. Any soft key can reference another one, but once a total of 32 active definitions are being processed, any further ones will be ignored. This limit should be ample for all reasonable uses of Soft Keys, and acts as a safety check in case you accidentally set up certain kinds of erroneous definitions.

### Other Considerations When Using Soft Keys

If a key hasn't been defined, it'll be ignored when used. Anything that could cause text loss requires confirmation from the keyboard. A definition that's too long is ended automatically. A definition that ends in the middle might leave the editor in Command or Control mode, but that will be apparent on the screen, and whatever was happening can be completed from the keyboard; in fact, you might want to do that sort of thing on purpose, in some cases.

### Stopping an Endless Soft Key

It's possible to define one or more soft keys so that they refer to each other in a circle. When one of them is pressed, it will supply its definition, and then reference @/Another soft key that will supply its definition and then reference @? the first soft key again. If this happens, you can interrupt the "loop" by pressing the "BREAK" key, and then delete the unwanted text. For example, if <X> and <Y> were defined as follows:

```
X: <Y> abc <Y> <Q> Y: <X> def <X> <Q>
```

Then, when <X> is pressed, this endless sequence would be developed on the screen:

```
abcdefabcabcabcabcdefabcabcdefabcabcdefabcabc
```

That repeating series of characters would continue to be added to the text until memory was full or until you pressed the "BREAK" key to stop the Soft Key processing. The limit of 32 repetitions doesn't apply here because the soft keys will keep refreshing each other.

### Nested Soft Keys

"Nesting" means that something is contained or done within something else. The "outer" function begins, and then the "inner" one is performed, and then the "outer" one is continued. You can do that with Soft Keys, too: a Soft Key definition can include another Soft Key in the middle, not just at the end. The first portion of the first Soft Key will be performed, the second soft key will be performed, then the rest of the first one will be used.

Up to 32 soft key definitions can be active at one time when you nest, chain, or "loop". Any excess definitions will be ignored.

### A SOFT KEY TO CUSTOMIZE THE EDITOR

If you find yourself issuing certain commands to customize the Editor every time you start to use it, you can put those commands into a soft key. For example, if you always wanted "Search" to distinguish between upper and lower case ("CASE Y"), and a "GRID" line to appear on the 4th screen line, you could define key <X> to do it for you:

```
CMD=> key x
DEFINE KEY -X- BREAK ca y ENTER DN DN DN CONTROL g <Q>
```

("DN" stands for "down arrow.") The definition of key "X" would be the "BREAK" key to start the command mode; "ca y", to tell "SEARCH" to distinguish between upper and lower case; "ENTER", to end the "CASE" command; three "down arrows" to move the cursor to the 4th line; and then <G> to display a grid line on the 4th line. After testing the soft key to make sure it works properly, the soft key definitions can be saved to "AL/DEF" for later use. Thereafter, as soon as you start the Editor, you can simply press <X> to execute that definition.

### A SOFT KEY FOR CROSS-FILE CHANGES

When we explained how "Search and Replace" work, we promised to show you how to do cross-file global changes safely, with only 3 keystrokes per file. Well, here's that explanation. We'll also show you how to do cross file searches with only 2 keystrokes per file.

The method to be used assumes that you are working with a long document that has been written as a series of appended files (not imbedded files). To set up an inter-file Search and Replace operation takes 3 steps, which don't count towards those 3 keystrokes, because they're only done once:

1. Define a Search argument with the "Search" command:

Example: CMD=> /disc

2. Define a Replace argument with the "Replace" command:

Example: CMD=> R disk

3. Define a Soft Key that will issue 2 commands: "SR" and "LINK":

Example: CMD=> KEY X

**<X>: BREAK sr\* ENTER BREAK link <Q>**

Now for the three keystrokes:

1. Press **<X>** (that's Control and "X"). The global search and replace will be done, but before the "LINK" is executed, the number of changes made will be displayed;
2. Press "ENTER" in response to the "change" message;
3. The "LINK" will be executed, since the Soft Key was only suspended, not cancelled, by the information message. Since a change was made to the file, the "LINK" will cause the Editor to ask you whether to SAVE, QUIT, or return to editing;
4. Press "S" to save the updated file. After it has been saved, the next file in the "append" chain will be read into memory;
5. To repeat the global change and initiate the next link, just press **<X>** again. In other words, by repeating steps 1 through 4, the entire chain of files can be processed. If you change your mind at any point, just press "BREAK" instead of "S", or don't use **<X>** after another file is brought into memory.

To summarize, the 3 keys are: **<X>**, "ENTER", and "S".

If you want to perform an inter-file Search, you don't even have to define a soft key: Once the Search argument has been defined, you can press **<F>** to repeat it, and **<&>** to do a "LINK" to the next forward file.

### Why 22 Soft Keys, Why 22 Keystrokes Per Key

The purpose of Soft Keys is to provide a simple, quick way of getting a lot of keystrokes into the computer, so making them work like normal control key functions seemed to be the best approach we could offer: only two keystrokes are needed for many of the keys, and 3 for the others. You may have noticed, by the way, that all the default definitions that would exit from the editor were placed in Shifted soft keys, to make it harder to press them by accident.

Most of the letters on the keyboard were already assigned to normal control key functions, so using the numeric row of keys for Soft keys allowed us to group them logically. That gave only 10 keys, so we used the Shifted numbers to give 9 more (SHIFT ZERO acts as the upper / lower case toggle, and can't be used for anything else). Finally, we wanted a way of allowing some repeating soft keys, and that led to the "X", "Y", and "Z" commands ("X" and "Y" were also in our other word processor, "NEWSCRIPT", as well as in a popular mainframe editor). Hence, the total of 22 soft keys.

22 keys of 22 characters each (plus one used by the editor) just about fill 2 disk sectors. The limit could have been made larger by reserving more memory for the definitions, but that would have reduced the size of the area reserved for normal text. Since soft keys can be chained together, "22" isn't really a limit anyway, so there seemed to be enough keys and keystrokes for most purposes.

\* \* \*

This completes the explanation of how soft keys can be defined, saved to disk, and loaded in from disk. It also completes the last features for the Text Editor, so if you've been reading right through this book, this would be a good place to take a break. When we pick up again in the next chapter, we will explain "everything there is to know" about Text Formatting (well, almost everything).

**CHAPTER 5**  
**THE TEXT FORMATTING LANGUAGE**

ALWRITE has many text formatting capabilities. You can learn just a few of them quickly, and still produce perfect, professional appearing letters. Those simple capabilities were presented back in Chapter 3. Other things were done for you automatically, by default: that's why ALLWRITE can be easy for the beginner to learn, and yet endlessly capable for the demanding user with sophisticated or special-purpose formatting requirements.

You can control the way a document is formatted by using the "Control Words" and "Emphasis Marks" that comprise ALLWRITE's text formatting language. (In NEWSCRIPT, "emphasis marks" were called "escape sequences", but that sounded too much like computer talk, so we changed it to English in this book.) If you've used other word processors in the past, it's likely that you're already familiar with the general method used here. Even if you haven't used that sort of thing on a computer, you've had to mark up your papers to indicate corrections, revisions, and changes. For example, the " ", a backwards facing double "p", is commonly used to indicate where a new paragraph should begin. That's why ",PP" is used by ALLWRITE; it's basically the symbol you would have used with pencil and paper.

Control words and emphasis marks are typed into the editor along with the rest of your text, while you're writing or revising your document. Soft keys can be used for the most common ones, such as "new paragraph", which is pre-defined as Soft key 1. When the Text Formatter runs, it processes your file(s) automatically, non-stop, and interprets the control words and emphasis marks as it encounters them.

**Control Word Requirements**

Control words must be placed on screen lines of their own. The line above a control word line must end with an "ENTER" symbol (the backward facing "L"), and the control word line itself also must end with the "ENTER" symbol. The first character of the control word line must be the control word symbol, which normally is the semi-colon. The next two characters must be a valid pair of letters, so that the three character sequence, ";xx" forms a recognizable control word to ALLWRITE. (";XX" is not a valid control word, by the way.)

Some control words can be followed by "parameters" which are words and/or numbers that provide additional information relevant to that control word. For example, ";TM 8" would set the Top Margin to be 8 lines high, and the text would start on the ninth line of each page. "8" is a parameter in that example.

More than one control word usually can be typed on a single control word line. Each control word must start with a semi-colon. The additional semi-colons and control words can start right after the end of the previous control word (or parameter), or you may leave spaces to make them more readable. All three of the following examples do exactly the same thing:

Example 1: ;tm 4  
;lm 5  
;ll 60  
;bm 6

Example 2: ;tm4;lm5;ll60;bm6

Example 3: ;tm 4 ;lm 5 ;ll 60 ;bm 6

The four control words used in these examples define the top, left, right, and bottom margins, respectively. Margins will be covered later in this chapter, after we finish explaining the ground rules for using them..

Certain control words must be the last ones on a control word line because their parameters could include things that look like control words. A few control words must be all by themselves on a control word line. ";BA" and ";AP" are the two examples of this shown in the previous chapter. The summary shown in Chapter 8 shows which control words must be the "last" or "only", and which ones cause "control breaks" to start new print lines.

### **Breaks in Text**

A "break" in text is something that causes the Text Formatter to stop printing one word after another, even when there's more room left on the print line. Four things can cause a break; reaching the end of the last text file to be printed; certain control words themselves; two "ENTER" symbols in a row; and one "ENTER" symbol when Formatting or Connecting is turned off (";FO OFF" or ";CO OFF").

Two "ENTER" symbols in a row cause a break and a blank line, just as they do on the screen. Certain control words cause a break in the text because they change the print setup or because you've used them to create a break. The simplest of these control words is ";BR" itself. If used, the text on the next screen line will be printed starting on a new print line. Other control words, such as ";PP" also cause a break because their functions imply a need to start a new print line. Some control words, such as ";IX" (to define a word that should go into the index) do not cause a break because they shouldn't: an index reference occurs in the middle of a paragraph, and the rest of the paragraph will be printed starting with the very next word, not as a new line.

As a rule of thumb, most control words will cause a break in the text; we will identify the ones that don't as they are covered. Please note that Emphasis Marks (explained next) never cause a break in the text.

### **Emphasis Marks Requirements**

Emphasis marks are used to control things like underlining and boldface. Since that kind of emphasis usually applies to a word or two, the marks are included in normal text lines. For example, we've been underlining and boldfacing our section headings throughout this book by doing this:

**@ \$ @ \* 2Emphasis Mark Requirements @ % @ \*0**

"@ \$" is the emphasis mark that starts underlining (Control 4), "@ \* 2" starts boldface level 2 (two overstrikes on a Diablo printer, "Double-strike" on an Epson), "@ %" turns off underlining (CLEAR 5), and "@ \* 0" (CLEAR 7) turns off boldface.

### **CONTROL WORD DEFAULTS**

ALLWRITE uses default values for all the formatting features it supports, and if those defaults are acceptable to you, you won't need to include the corresponding control words. The most important of these defaults are that the margins are pre-set to one inch each, resulting in a printed line length of 6 1/2 inches; and that text is left and right justified to produce smooth margins. The default pitch is either "10" or proportional, depending on your printer; no emphasis or darkness is used by default; and the pages are numbered in the upper right hand corner, beginning with page 2 (we don't think it's a good idea to number a one-page letter).

Of everything just mentioned, the only thing of real importance is the automatic justification, because there are two aspects to it. One is the smooth margins, and the other is that all text from the file is treated as a continuous series of words to be printed in a single paragraph. This paragraph will end when a control word line is encountered, or

when two "ENTER" symbols in a row are found. Otherwise, regardless of how the text appeared on the screen, it'll be printed in such a way that as much as possible will be fitted onto each print line, and spaces or micro-spaces will be added evenly to make the right margin smooth and straight.

Since most letters, papers, and books need occasional "breaks" in the text, new paragraphs, blank lines, centering, etc., you'll want to use some of the control words and emphasis marks to guide the Formatter to give you the results you want. The ways to do that will be the subject of the rest of this chapter.

#### **The Printer Defaults File -- xxx/DEF**

If you had to specify all the details of your printer and your normal requirements each time you wanted to use ALLWRITE, you'd never even get started. So, we've designed ALLWRITE to use several "default" files, and we've set these up in advance for you. A lot of this was done by the installation program, and the rest was contained in the various files we asked you to copy to your working disk.

The Installation Procedure created two small files for each printer you selected. The extension on one of these was "/TAB" and the other, "/DEF". For example, if you selected an Epson MX-80 with Graftrax Plus, and a Diablo, the four files would be:

**MX80GP/TAB MX80GP/DEF DIABLO/TAB DIABLO/DEF**

You cannot modify the "/TAB" files, and any attempt to do so will cause ALLWRITE to "crash" when you try to print. However, you can modify and tailor the "/DEF" files to match your preferences, since they are just lists of ALLWRITE control words. You can set the margins, pitch, darkness, and just about anything else in those files, and have a different setup for each of your printers. Before doing so, we suggest you print a few things to see how they look with the supplied defaults.

Please note that "AL/DEF" is not a list of control words; it cannot be modified by the editor, but only by the Installation Program or the "KEY SAVE" command.

#### **Emphasis Marks**

ALLWRITE recognizes twelve emphasis marks, most of which occur in pairs (to turn something on or off). Some of these marks don't apply to certain printers, and are ignored if used with them. For example, "double-width" won't work on a daisywheel printer, and "super-script" works only on printers that have either super-scripts (GRAFTRAX-PLUS) or reverse half line feed. As we've said in our catalogs and our ads, "certain features are printer dependant," which is a nice way of saying that if your printer can't do it in the first place, ALLWRITE cannot overcome the missing feature.

These are the emphasis marks used by ALLWRITE:

- @\$ turns on underlining, works on most printers
- @% turns off underlining
  
- @\*1 turns on boldface, works on most printers
  - any number from "1" through "9" can be used.
  - 1 = overstrike, or "Emphasized" on Epsons, etc
  - 2 = two overstrikes, or "Double-strike" on Epsons
  - 3 = three overstrikes, or "Double Emphasis" on Epsons
  - 4-8 = that many overstrikes, depending on the printer
  - 9 = shadow print on some Daisywheel printers
  
- \*0 turns off boldface, regardless of previous level

- (@ turns on double width (dot matrix only)
- (@) turns off double width
- @/ turns on italics on Epsons and some others, uses red ribbon on Diablo, if present
- @? turns off italics, returns to black on Diablos
- @+ begins a super-script or reverse half-line feed only if your printer can do it
- it cancels sub-scripts on half-line feed printers, prints half-height characters with GRAFTRAX-PLUS
- @- begins sub-script or forward half-line feed only if your printer can do it
- It cancels super-scripts on half-line feed printers, but only switches to sub-scripts on Epsons
- @= cancels sub and super-scripts on Epsons, etc. It has no effect on half-line feed printers.
- @< backspaces one character to do an overstrike, works only if your printer can do backspace
- @> signals that printing should pause until the "ENTER" key is pressed. Useful for wheel or ribbon changes on some printers.

If you forgot to turn off certain emphasis features, they will produce unusable documents. For example, if underlining is set and left on, the rest of the document will be underlined. To catch this, we suggest you make it a practice to preview your printing with the ";VI" run-time option when starting the Text Formatter. Once you know a section of the document is clean, you won't have to check it again.

### **PRINTER LIMITATIONS**

You can save some money by reading this section. One of the most frequent phone calls we get on "NEWSCRIPT" is that its boldface capability suddenly stopped working on an EPSON printer. Of course, that can't happen with software, and what actually happened was that the caller tried to use boldface with super-scripts, sub-scripts, 12 pitch, or 16 pitch. Unfortunately, the EPSON doesn't allow "Emphasized printing" (Our Boldface Level one) in combination with those features (neither do the GEMINI or other printers similar to the EPSON), so it rejects part of your request and honors the rest.

These restrictions usually are documented in the printer manuals, but often aren't shown too prominently (who wants to talk about a limitation) So, when you find something along these lines that doesn't work, try to remember what we just said, and check the printer book before calling us. We take all problem reports seriously, and will be glad to explain the problem if you do call, but neither you nor we are in business for the purpose of enriching the phone company.

### **RUNNING THE TEXT FORMATTER -- SOFT KEY <!> or <">**

Some of this material was discussed near the end of Chapter 3, the introductory tutorial. It's included again because this chapter concentrates on all aspects of the Formatter.

Before using the text formatter, you must create a file containing the text you want to print and the control words that describe the layout. You do this with the editor, and then store the file onto a disk to preserve it. Some word processors combine their editing and formatting functions into a single program, but the limited memory of the TRS-80 and the range of capabilities of ALLWRITE makes that impractical: If the two programs were combined, you would be able to edit only one page at a time, and the disks would be running so often that you'd get very little work done.

Once you've written something, the Text Formatter can be started from the Editor by pressing Soft Key Capital 1 (automatic, no prompting operation), or Soft Key Capital 2 (which allows you to select a different text file and to specify some run-time options to override the normal setup). You could use the Editor's "RUN ALF/CMD"

instead of a soft key, but that would mean more typing. You can't really run the Formatter directly from DOS, because, even though it'll partly run, it won't know which printer you're using: the Editor has to supply that information, along with some other things we've taken care of for you.

In automatic mode, started by <!>, you don't need any operating instructions, because there's nothing for you to do. The document will be formatted and printed, then control will return to the Editor.

In prompting mode, you'll be asked which file should be printed, and the name of the file you were just editing will be shown as the default. If that's acceptable, press "ENTER", otherwise, type in another name, or press "?" to display a directory. If the file isn't found, you'll be asked to supply another name. If you want to cancel the Formatter, reply with a dash "-". When requesting a directory, a drive number and a mask can be used, just as with the Editor.

### Run-Time Options

Once the file has been found, you'll be asked to enter any one-time, run-time options. If you're happy with the normal setup, just press "ENTER" and printing will begin. If you want to change the setup, you can type a control word line, beginning with a semi-colon and containing any valid control words.

:pp

Almost every control word in ALLWRITE may be used as a run-time option (unlike NEWSCRIPT), but some of them are most useful as run-time options, while others are only useful in normal text. Some commonly-used run-time options are shown below, with the control word portion capitalized and underlined, and the rest of the word, which you don't type, in lower-case.

**;Video ;Edit ;Smy ;PGrange**

After you've typed the options list, press the "ENTER" key, and printing or previewing will begin.

### Pausing the Formatter

You can suspend printing or previewing at any time by pressing SHIFT "@", and resume by pressing "ENTER". You can interrupt printing or previewing for the purpose of cancelling it or activating "Mini-Edit" by pressing "BREAK". If you do this, you will be asked:

**DO YOU WISH TO CONTINUE PROCESSING (Y/N/E, DEFAULT= Y)**

"Y" means "Yes, continue: I didn't mean to press BREAK". "N" means "No, cancel the rest of the printing and display the Exit menu." The page cannot be ejected automatically if you make this choice. And, @/(but see the "T" option of the Exit menu below).@? "E" means "Activate 'Mini-Edit' so I can fix something before it's too late".

"Mini-Edit" will be explained in a moment. It's a way of adding, deleting, or changing the text after it's been read from the disk, but before it gets formatted or printed. Your changes cannot be saved to disk, but you can modify anything in the file on a temporary basis.

### Exiting from the Formatter

When printing or previewing ends normally, you will be asked whether the final page should be ejected:

**EJECT PAGE (Y,N,C DEFAULT = Y)**

If you were running in automatic mode, the page is just ejected. "C" means "continue", and allows you to type some extra text and control words directly from the keyboard. If you cancel printing, this question is not asked, and the paper is not ejected.

**Exit Menu from Text Formatter**

When running in "automatic" mode (Soft Key Capital 1 from the Editor), the Text Formatter just returns control to the Editor. When running in "prompt" mode, you will be offered these choices:

**PRESS P TO PRINT AGAIN, E TO EDIT, OR Q TO QUIT**

The default is "P"; "E" returns control to the Editor, while "Q" exits from ALLWRITE to DOS. If you choose "P", you can select a different file for printing, or re-print the same file, possibly with different options.

**PRINTER STATUS AFTER FORMATTING**

When printing ends, the printer is left in whatever pitch had been in use. This is useful if you don't know how to send internal feature selection codes to the printer from BASIC. To totally reset the printer, just turn it off and back on.

**CUT SHEET PRINTERS -- CS Control Word**

Most printers are used with continuous-form paper, which is either fan-fold or on a roll. Some printers can use individual sheets of paper, like a typewriter. Those individual pages are called "cut sheets", and must be fed into the printer by hand. If you have an automatic sheet feeder, this discussion probably doesn't apply to your printer, since it'll work as though it was using continuous forms.

If you will always be using cut sheets, you can specify that during the Installation procedure, and the Formatter will stop at the end of each page and prompt you to press "ENTER" after you've changed the paper. However, if you usually will use continuous forms, and will use cut sheets only occasionally, you should install for continuous forms. Then, when you want to use cut sheets (the final printing on good paper or letterhead), you can specify this as a run-time option.

**;CS**

That stands for "Cut Sheet operation" and lasts for the rest of the printing of the current document. If you've used it in your printer defaults file, and want to override it as a run-time option, just add "NO" or "OFF" after it:

**";CS NO"**

**MINI-EDIT -- ED Control Word**

Although this feature can be very useful, it is not essential to using the Text Formatter, so you may wish to skip it for the time being. There will be times when you will want to make small changes in a text file while it's printing: you won't want a certain control word to be used this particular time; an error was detected in the text; you want to add something; or you just want to experiment with the Text Formatter without having to use the editor at all. All of these can be done by activating the "Mini-Edit" feature in the Formatter.

Mini-edit can be activated in any of five ways:

1. as a Run-Time option. Control word is: ;ED
2. by pressing "BREAK" and then "E" during formatting
3. when an error is found and ";ER E" is in effect (the ";ER" control word will be covered later on)
4. by typing a semi-colon when the Text Formatter asks for the name of the file to be printed. In this case, all text must be entered from the keyboard.
5. by replying "C" (continue) when asked whether or not the last page should be ejected.

Before going further, we must emphasize that Mini-Edit is only useful as a way of changing things at print time. What you type cannot be saved to disk, only printed. Its purposes are to give you a way to do last-minute touch-up, and to have a very fancy typewriter for experimentation.

When Mini-edit is active, you will be prompted to enter text from the keyboard. There are two different prompts. The first one is used to allow you to modify text as it is read from disk, and the second is used when a disk file is not in use, or has finished printing (a P.S., for instance). These are the two prompts:

**\*EDIT\***                    **\*ADD\***

"\*EDIT\*" is the prompt when Mini-Edit was activated by methods 1, 2, or 3. (above); "\*ADD\*" is the prompt when it was activated by methods 4 or 5.

When the prompt is "\*ADD\*", just type whatever you want into the computer. Your only means of correcting errors is to use the left arrow to backspace over them and retype correctly, and this must be done before the "ENTER" key is pressed. You may type control words or text. When you want to stop, just press "ENTER" without typing any text: you'll be asked to confirm that you really are done. Up to three screen lines at a time can be typed before you press "ENTER".

When the prompt is "\*EDIT\*", a line of text or control words from the disk will have been displayed but not yet processed. You may reply in any of several ways:

ENTER	print text as-is
D	Delete displayed text
I newtext	Insert "newtext" ahead of current text
R newtext	Replace displayed text by "newtext"
T	Terminate mini-edit and continue printing
E	End printing, go to Formatter Exit menu

(A "mnemonic" for these is "TIRED".) Note that when Inserting or Replacing text, the letter "I" or "R", followed by one space, must be included. All text after that space will be used for insertion or replacement. The maximum length of "newtext" is three or four screen lines (the cursor will just stop when the limit is reached), but you can insert more text on the next prompt.

If the text or control words from the file are acceptable to you, just press "ENTER" and they will be processed. "Delete" does not ask for confirmation (it doesn't change the disk file, so nothing can be lost). "Insert" will display the same text again after processing what you've typed. "Terminate" is the normal way to back out gracefully and let the Formatter continue processing the rest of the disk file automatically. (You can always press "BREAK" later on to reactivate Mini-Edit.) "Exit" is used when you decide the situation is beyond repair, and it displays the Exit menu immediately. You can restart the same document or return to the editor for more extensive, permanent changes.

### **ORGANIZATION OF THIS CHAPTER**

The rest of this chapter explains all the formatting features of ALLWRITE. The features are organized by topic, and the topics are listed below. Each feature is indexed in several ways at the end of the book for easy cross-reference, and most of them can also be found on the "HELP" menu of the Editor.

### **TOPICS IN THIS CHAPTER**

- margins
- hyphenation
- indentation, hanging indents, and centering
- paragraphs and blank lines
- numbered paragraphs
- changing control word and emphasis symbols
- double spacing, lines per inch
- paper size
- justified and as-is printing
- pagination, page numbering, and anti-widowing
- printing selected pages, starting page number
- cut sheets (non-continuous pager)
- headings and titles
- pitch, proportional and micro-spacing, print wheel tables
- character translation and special symbols
- turning off certain features temporarily
- darkness and underlining -- another way
- comments in text
- modifying text just before it prints
- pausing for disk change
- multiple (carbon) copies
- multiple columns
- error control
- form letters
- mailing labels
- legal documents
- tables of contents
- indices
- footnotes
- variables and logic within text

**MARGINS**

Margins form the borders around a printed page. ALLWRITE lets you set all four of these, and you can change them as often as necessary throughout a document. Besides the outer margins, there are four others that can be controlled, if necessary. This diagram illustrates most of the margins, and all of them will be explained afterwards:

	1 2       ;TM 6, 1 (TOP MARGIN) 3 4 5       < - page number/title here -> 6	
:LM 5	(LEFT MARGIN = $\frac{1}{2}$ inch)  < - - - - - ;LL 65 - - - - - >  ( LINE LENGTH = 6 $\frac{1}{2}$ inches )	
:OM 5 ;EM 5	(ODD PAGE MARGIN) (EVEN PAGE MARGIN)	

**:TM 6,1**

Top Margin in lines. This defines the space from the top edge of the paper to the first "body" line, where normal text begins to print. Titles and headings that repeat from page to page are printed within the top margin and do not affect its size. The top margin must be large enough to hold all the titles: usually, you'll only use one title line, but you can define up to 9 top titles (and 9 bottom ones), so it might be necessary to increase this number. The top margin cannot be set to zero if any top title definitions are in effect.

By default, the top margin contains "6" lines, which corresponds to one inch if you print 6 lines to the inch. A secondary margin is also definable with ";TM", and sets the number of blank lines to be placed between the last title line and the first body line. By default, this value is "1", so there normally will be four blank lines, a title line, and a blank line above the text on each page. When printing this book we set that secondary top margin to "2", in order to keep the title well-separated from the body.

**:BM 6,1**

Bottom margin, in lines. This defines the space from the last body line to the bottom edge of the paper. Everything else about it is the same as for the Top Margin, but the top and bottom margins are treated completely independently of each other.

**:LM 5** Left margin, in tenths of an inch. This defines the space from where the printer can start printing to the place where it will start. Most dot matrix printers can't print closer than about a 1/2 inch from the left edge, so the default of another 1/2 inch is intended to produce a one inch left margin. You may need to change this value so it matches your printer. If you do so, put the new value in the defaults file ("xxx/DEF") for your printer.

**:OM, :EM** These let you control the odd page and even page left margins separately. ";LM" does both at once.

Note to former NEWSCRIPT users: "LM" has replaced "AD", which is no longer supported.

**:LL 65** Line length, in tenths of an inch. This defines the maximum width of a body line, starting just after the left margin. ALLWRITE doesn't have a ";RM" (right margin) control word, because ";LL" does the same thing, and ";RM" is ambiguous: does it specify the width of the right margin, or the starting column, working out from the left edge? At any rate, the combination of left margin, line length, and printer starting position are intended to leave about one inch for the right margin. These are the normal sizes for most letters and papers, but you can change them in each document or in the defaults file.

### **Line Length and Right Margin -- LL Control Word**

You can change Margins over and over again, even within the same page. The top and bottom margins generally don't change, but you might want to indent a quotation on both sides. We will cover left-side indentation later on, but right-side indentation is controlled by shortening the line length. This control word can accept a number, a signed number, or no number at all:

- :LL n** 'n' sets the new permanent line length. When multiple columns are in effect, this changes the column line length, but not the overall line length.
- :LL -n** a negative number sets a shorter, temporary length.
- :LL +n** s signed positive number sets a longer, temporary length. This new length can be bigger than the permanent one unless multiple columns are in effect.
- :LL** no number at all restores the permanent length.

### **Using Letterhead Paper -- LG Control Word.**

If you're preparing a letter that will be several pages long, and want the first page printed on your pre-printed letterhead, a problem arises: the top margin on that first page must be different from the top margin on the remaining, blank pages. The solution is to use the "LoGospace" control word:

**:LG n**

where "n" is the number of lines from the top edge of the paper to where you want the printing to begin. When you use this control word, you should position the first page so that the printhead is already on the first line to be printed. ALLWRITE will ignore the normal top margin on the first page when ";LG" is present, but use only the normal top margin for all subsequent pages.

";LG" almost always will be used in conjunction with the "CUT SHEET" (;CS) control word or with printers pre-installed for cut sheet operation. However, for scratch copies of a letter, you can position the first page of continuous form paper to the middle, and let printing run continuously.

### Footnote Margin

This is the eighth kind of margin, and will be covered under the "Footnotes" topic near the end of this chapter. It's only useful when you use footnotes.

To summarize the eight margins: top; secondary top, bottom, secondary bottom, left, line length (right), logo space, and footnote.

### HYPHENATION -- HY Control Word

We'll cover this early because ALLWRITE makes it very easy to hyphenate your text. You will notice, however, that we didn't use many hyphens in this book: with proportional printing, hyphenation really isn't necessary, and splitting words makes them harder to read. Hyphenation is enabled by default:

**;HY c [, ON or OFF]**

"c" is a character that will be used to mark conditional (soft) hyphenation points. "ON" is the default, and tells "ALLWRITE" to use these soft hyphens as well as normal dashes. "OFF" tells ALLWRITE to throw the soft hyphens away and not attempt hyphenation. By default, the soft hyphen character is the shifted "AT-sign", which appears on the screen as a reversed apostrophe or as an English "pound" (money) sign.

Once the soft hyphens are in the text, the Formatter will take care of the rest. If text shifts around, you won't need to change these marks, and you won't be prompted during printing to help the Formatter split up the lines. So, the only question is how to get soft hyphens into the text.

There are two ways to do it: right from the keyboard, by pressing the shifted "@" while typing or inserting it afterwards; or the easy way, by using ELECTRIC WEBSTER's Hyphenation Feature, which does it for you automatically while also checking your spelling. The advantage to the first method is that it's free, and if you don't expect to use hyphenation very often, it's the right approach for you. The advantage to Electric Webster's option is that it does the whole job for you, quickly and painlessly.

#### Example hyphen'ation im'proves the ap'pear'ance

(We used a normal apostrophe in the example because our print wheel doesn't have a reversed one.) If a printed line ended in the middle of any of those words, the nearest "" would be changed to a "-" and the rest of the word moved to the next print line. All the unused "" symbols would be removed from the text, as though they had never existed.

### INDENTATION

"Indentation" is a way of temporarily moving the left margin further to the right. Because ALLWRITE allows you to mix pitches on the same page while keeping the margins straight, all indents are measured in tenths of an inch, rather than in character columns. So, an indentation of "5" increases the left margin by half an inch. In normal 10-pitch, that would leave 5 spaces to the left of text, and the text would start in the sixth character position. The same indentation in 12-pitch would leave 6 spaces.

It's also possible to temporarily move the right margin to the left by changing the line length (;LL"), as explained earlier. Left margin indentation can be done in seven different ways in ALLWRITE, depending on the situation of the moment:

:IN n	indent "n" tenths of an inch from the left margin. This usually is the best way for simple things.
:HI n	automatic hanging (delayed) indent
:OF n	semi-automatic hanging (delayed) indent
:CE n	centering, which is different from indentation
:TB	tabbing, as explained in Chapter 4
:PP	new paragraph indents first line. Covered later.
:LM n	change the left margin, as explained earlier. This is the worst way.

This section will explain the first four of these. Tabbing was explained in the previous chapter, along with on-screen tabbing. Using paragraph controls will be the subject of the next major topic in this chapter.

Changing the left margin was explained in the previous topic. It is not a good way to do indentation because the size of the left margin for the entire document may have to change if you move the tractors that guide paper in your printer, and it would be very tedious to have to go through a long document to adjust all the left margin settings afterwards. The right way to do this is through the ";IN" control word.

### Simple Indentation -- IN Control Word

The "INDENT" control word moves the left margin temporarily to the right or left, but never further right than the current line length, and never further left than to the permanent left margin set by ";LM" or by default. The "INDENT" control word can be used in any of four ways:

:IN n	indents "n" tenths of an inch past left margin Example: ;in3 indents 3/10 of an inch
:IN +n	adds "n" to current indent Example: ;in +5 indents another 1/2 inch.
:IN -n	subtracts "n" from current indent. If result is less than zero, stops at zero (left margin) Example: ;in -5 balances previous +5
:IN	if no value given, indentation ends Example: ;in goes back to left margin

Simple indentation can be used by itself, or in conjunction with hanging indents and tabs, to produce good-looking lists, or with relative line lengths to produce narrow paragraphs, such as these:

Proper English usage requires that all quotations more than 3 lines in length be shown indented, as separate paragraphs. Shorter quotations may be included in a paragraph of normal text.

That was done as follows:

```
;sk;in+5;ll-5
Proper English ... normal text
;sk;in;ll
```

Relative indents (;IN +5) are useful when you want to keep changing the indent back and forth without having to keep track of the actual current position. They also make it easier to add another indent level later on, without having to re-calculate all the existing ones. ALLWRITE will stop negative relative indents when they get back to the left margin, you don't need to worry about overshooting.

### Hanging (Delayed) Indents - - HI and OF Control Words

```
;HI n or +n or -n or 0
;OF n or +n or -n or 0
```

A "hanging indent" is a paragraph in which all lines after the first are indented, so that the first line extends (hangs) out to the left of the rest. We've been using this format frequently throughout this book, and it's easier to illustrate than to explain it. For example:

- Hanging indents let you list a series of points in such a way that they are kept separate and easy to read;
- To insure a perfect alignment of the indented material, a tab mark should be placed between the part that hangs out (the asterisk in this example and the first word of normal text);
- The tab should be selected so that it matches the size of the delayed indent. Rules for this are given below;
- A normal indent and a temporary shorter line length were used in this example to make sure the points stand out as a list.

Here's the control word line used in the above example:

**;sk;in5;ll-5;tb+9;hi3**

We indented both margins (;IN5;ll-5) a half inch, set a tab stop at "9", which is 8/10ths of an inch past the left margin (tabbing ignores indentation), and set the hanging indent (;hi3) to indent 3/10ths of an inch on all lines except the first. The tab stop was needed to make sure the text of the first line would be aligned with the automatically indented text of subsequent lines: tabbing starts at column 1, while indentation starts at column zero.

There are two kinds of hanging indents: automatic and semi-automatic. ";HI" is used for automatic hanging indents. These restart every time they encounter a control word that causes a "break" in the concatenation of text. If your list will have just one paragraph per point, automatic hanging indents are the ones to use, because they're so easy. Like a normal ";IN" control word, the indentation for ";HI" can be an absolute number or a relative one. Since hanging indentation continues until turned off, you must remember to place an ";HI" after the last line of indented text.

If you want to have a couple of paragraphs within a single hanging indent, don't use ";HI", because it'll hang out the first line of the second paragraph. Instead, you can use ";OF", which stands for "OFFset", and which works just like ";HI", except that it only hangs out the line that follows it. All following lines will be indented, even across text breaks, until another ";OF" is encountered. Like ";HI", it must be used with a number, and when done, it must be turned off by ";OF 0". In the example shown earlier, we used ";HI3" and separated each point by a simple ";BR" (break). If we had used ";OF3", we would have had to repeat it between each paragraph. Since hanging indents cause a break, the ";OF3" could have been used in place of the ";BR": it would have caused its own break, and also made the next line of text hang out to the left.

### **Hanging Indents with Tabs**

Hanging indents (and offsets) are expressed in tenths of an inch, regardless of the pitch you're using. Tabs normally are expressed in tenths of an inch also. If you want to use them together, there are just 3 things to consider:

1. A hanging indent sets the number of tenths of an inch that should precede the text, whereas a tab sets a starting position for text. This means they differ by "1": a hanging indent of "3" will line up with a tab stop of "4".
2. A hanging indent can be preceded by the space defined for an indentation (as we are doing in this paragraph), which makes the total indent cumulative. However, tabs are always set relative to the left margin, and are not affected by indentation. This means that, to line up a tab with an indent and a hanging indent, you should mentally add up the two indents.
3. The optional "C", used in a tab definition to force use of the current pitch, should not be used.

These three rules boil down to this: if you want to line up an indented series of numbered paragraphs like the ones above, the tab stop should be the sum of the indent, plus the hanging indent, plus one:

**;in3;hi3;tb>7**

That's what we used above. We also used "LL-3;PT", if you're wondering about the right margin indentation and the numbering.

### **CENTERING - - CE Control Word**

In Chapter 3, we showed you how to center a single line of text. Actually, you can center several lines at a time:

<b>;CE</b>	centers the next one line of text
<b>;CE n</b>	centers the next "n" lines of text
<b>;CE ON</b>	centers all following lines of text until centering is turned off
<b>;CE OFF</b>	turns off centering

The margins, indent, and hanging indent all are considered when centering text. In monospace (10, 12, 16 pitch), it is possible for centering to be off-balance by one character, if the effective line length is even and the text length is odd (or vice-versa). In proportional pitch, centering can be off by one printer unit, but since these are 1/60th, 1/120th, or less of an inch, it is generally undetectable.

### **VERTICAL CENTERING -- VC Control Word**

This generally is used with one-page letters or title pages. All the text for the page accumulates in memory, and then ALLWRITE centers it between the top and bottom margins before printing it. If some text has already been printed on the page before ";VC" is issued, the rest of the text will be centered between the previous text and the bottom margin. The top and bottom margins are specified with the ";TM" and ";BM" control words.

";VC", like many other control words, takes physical page size into account. If you are using normal, eleven inch paper, you should always specify ";PL66" or no page length at all (66 is the default): do not try to "help" the Formatter by second-guessing it with a false page length.

### **BLANK LINES -- SK Control Word**

There are several ways to leave blank lines in printing: you can press "ENTER" more than once, and each extra "ENTER" will create a blank line on paper; you can specify the number of blank lines that should precede each paragraph (see the next topic); or you can just tell the Formatter to leave some blank lines by using this control word:

**;SK [ n [ ,A ]]**

Example: **;SK 2** leaves 2 blank lines

"n" is the number of lines to be skipped. If omitted, it defaults to '1'. The number can be negative, in which case reverse paper feeding will occur on printers having that capability. The number can include a ".5" to indicate a half-line feed should be done on printers having that capability. ";SK 1.5" would leave one and a half blank lines between the previous printed line and the next one. ".5" is the only allowable decimal fraction; you can't use "2.75" or anything else along those lines.

If the "skip" occurs near the bottom of the page, the extra lines not yet skipped will be discarded, and printing will resume on the very first body line of the next page, unless the "A" (always) parameter is included. If "A" is specified and there is not enough room on the current page for all the blank lines, they will be skipped on the top of the next page.

If the "skip" occurs at the top of the page, it will be ignored unless an "A" (always) is used:

<b>;sk 3</b>	will be ignored at top of page
<b>;sk 3A</b>	will be honored at top of page

### Blank Screen Lines

If you have an entirely blank line on the screen, one that doesn't even have an "ENTER" on it, that line will disappear when it's scrolled off the screen, or when lines are joined together. Totally blank screen lines cannot be saved, because they don't even exist, as far as ALLWRITE is concerned. So, if you want blank screen lines to be preserved, put an "ENTER" symbol on them.

### PARAGRAPHS -- PP Control Word

There are two ways to start new paragraphs in ALLWRITE. One of them is to press "ENTER" twice when at the end of a line, and then to space or tab over to where you want printing to begin on the first line of a new paragraph. That's how it's done on a typewriter, but a much better way is to use Soft Key 1 or the "paragraph" (;PP) control word.

**;PP [ n ] [ ,b ] [ D ]**

Example:      ;pp

"n" is the indentation for the first line of the paragraph, in tenths of an inch, and it defaults to "5" (1/2 inch). "b" is the number of blank lines to leave between paragraphs, and it defaults to "1". It may be a decimal number if you want some form of half-line spacing, such as "1.5". "D" is the letter "D" for "Delay", and its use is explained below.

Once ";PP" has been used with the values you want, or if you like the defaults, you don't have to specify the values on subsequent ";PP"s in the rest of the document. The most recent explicit settings will be remembered:

```
;pp4,.5
It was a dark stormy night ...
...
;pp
Pearched atop a craggy mountain ...
...
```

The first line of every paragraph will be indented four tenths of an inch, and an extra half-line will be skipped above each paragraph.

Since new paragraphs are one of the most common of all formats, we've defined Soft Key 1 <1> as the paragraph key for the Editor. It doesn't specify any values, so the defaults in the "printer/DEF" file, or any explicit values in your document, will be used. Since the soft key is so quick and convenient, we recommend you get in the habit of using it when writing with ALLWRITE.

Like all control words, ";PP" must occur on a control word line. Normally, it affects the very next print line, but if you want to change its values in a defaults file, without causing it to take effect immediately, you can put a "D" after the numbers: the "D" stands for "Delay", and tells ALLWRITE to just store the values for future use:

Example: ;PP 4, .5 D

That will cause a first line indentation of 4/10ths of an inch, and that first line will be preceded by a 1/2 line feed (that only works if your printer can do half line feeds, of course). However, because the "D" was specified, no paper movement or printing will occur when this control word is processed. Its values will be stored and the Formatter will go on to the next thing in the file.

**NUMBERED POINTS OR PARAGRAPHS -- PT Control Word**

You've seen some numbered lists in this book. We didn't type those numbers ourselves: we let ALLWRITE do the numbering for us. The reason for doing so was not to save typing time while entering the lists, but to avoid having to renumber them when something had to be added, removed, or moved around in the lists. This is a nice feature, and it's very easy to use.

There are several kinds of numbering capabilities in ALLWRITE: Legal numbering (a topic near the end of this chapter), Figure numbering (explained under "LOGIC" at the end of this chapter), and point/paragraph numbering, which is what we're talking about here. The control word to use is:

**;PT [ n [ c ]]**      Example: ;PT 1

"n" is a starting number, usually "1", of course. If omitted, "1" is added to the previous value, and the result is printed. If you don't specify a number the first time you use ";PT" in a document, the number "1" will be printed. Once you've set the starting number, you won't need to supply any numbers the next time you use the control word, because ALLWRITE will sequentially number your list as it prints it. "c" cannot be the "control word indicator" (usually a semi-colon).

"c" stands for the character to be printed after the number. It's optional, the default is the period ".", and that's what you'll want to use most of the time. If you wanted to use a right parenthesis, you would initialize a list this way:

**;PT 1)**

A hard space is added after the character, completing the sequence. The ";PT" control word should appear on the control word line just before each new point or paragraph you want numbered. It causes a "control break", insuring that the next text will start on a new line.

Point numbering is often used with hanging indents, so a tab stop should be used as well. To illustrate what we mean, the material below was used to produce the numbered list in the middle of Chapter 3:

```

;sk;hi3;tb+4;pt1
+it's fairly easy to learn ("CE" means "center"; "pp"
means "new paragraph");
;sk;pt
+it's @ *2very@ *0 flexible and extensive: you can
produce just about any combination of formatting
results you want to, with ALLWRITE, and not have to
work very hard in the process;
;sk;pt
+the formatting markers are highly visible: the control
words appear on the screen along with normal text,
so you will always @2*know@*0 what setting is in
effect.

```

When printed, this is the result:

1. it's fairly easy to learn ("CE" means "center"; "pp" means "new paragraph");
2. it's very flexible and extensive: you can produce just about any combination of formatting results you want to, with ALLWRITE, and not have to work very hard in the process;
3. the formatting markers are highly visible: the control words appear on screen along with normal text, so you will always know what setting is in effect.

### Changing the Control Word and Emphasis symbols (or)

#### How We Printed The Example Above -- CW and ES Control Words

In case you're wondering how we managed to fool the Formatter into printing those control words and emphasis marks without messing anything up, there's really nothing to it. We changed the control word symbol to a period and the emphasis mark to an exclamation point by doing this:

**;ES!;CW.**

";ES" is the control word used to redefine the first character of all Emphasis marks. ("ES" is partly a holdover from NEWSCRIPT, and besides that, we couldn't use ";EM" because that defines the even left margin.) Whatever character follows it becomes the marker for these from that point on, replacing the default at-sign, "@".

";CW" is the control word used to redefine the Control Word symbol, which usually is the semi-colon ";". If you're used to using a different symbol (probably a period "." or greater than symbol ">"), and don't want to change over to ALLWRITE's defaults, you can just stick those new definitions in your printer defaults file, redefine the Soft Keys, and never have to think about it again.

By the way, when you redefine the control word symbol, the new character goes into effect instantly, so the very next control word will have to use it, even on the same line:

**;CW..ES** ( NOT: ;CW;ES )

There will be good reasons for re-defining the Emphasis Mark symbol from time to time, but the only reason for not using the semi-colon as the control word symbol is that you're used to something else. We chose these symbols to make typing them quicker and easier than their NEWSCRIPT counterparts.

#### Line Spacing -- LS and LI Control Words

Normally ALLWRITE prints everything single-spaced, six lines per inch. You can change these as often as necessary, within the limits of what your printer can do. If you want 1.5 spacing (if your printer can do it), double-spacing, or triple spacing, use this control word:

**;LS n n=1, 1.5, 2, 3, 4, ...**

"1.5" is the only allowable fractional value, and not all printers can do "1.5" spacing.

Example: **;LS 2** (starts double-spacing)

If you want a different number of lines per inch than "6", and if your printer can handle other increments, use this control word:

**;LI n n=6, 7, 8, or 12**

Example: **;LI 8** starts 8 line/inch printing

All printers support 6 lines per inch (lpi), and many others support 8 lpi. 12 lpi is pretty small, and is intended for use with the half-height characters available thru "super-scripts" on Epsons with GRAFTRAX-PLUS, or with 15 pitch print wheels on certain daisywheel printers. If you try to use 12 lpi with normal size characters, the lines will run together.

**;LI7** simulates the spacing used in many magazines and newspapers. On most printers that can approximate this kind of spacing, the actual results won't be quite "7", but will be pretty close. For example, Epsons can space in 1/72" or 1/216", and "7" does not divide evenly into either of those numbers. Diablos and Spinwriters can space in 1/48", and "7" doesn't divide evenly into that, either. ALLWRITE will come as close as possible, and will adjust its positioning at the end of each page so as to re-synchronize properly with the top edge of the paper: There shouldn't be any cumulative "drift".

The effects of line spacing and line height appear after the next line is printed (or spaced), so the spacing above the next print line will be the old spacing (the paper was already moved).

#### Allowable Combinations of Spacing

**;LS** and **;LI** can be used together in any combination, and different values for both can be intermixed on the same page: if your printer can perform the vertical movements you've requested, ALLWRITE will keep track of where it really is on the page, let you double space at 8 lines per inch, and still arrive correctly at the bottom edge of the paper.

#### Some Examples of Line Spacing

<b>;LI 8</b>	8 lines per inch
<b>;LI 6</b>	6 lines per inch
<b>;LI 7</b>	7 (or 7-1/7th) lines per inch
<b>;LS 1.5</b>	1-1/2 line spacing, 4 lines per inch
<b>;LS 2</b>	double spacing
<b>;LS 3</b>	triple spacing
<b>;LS 2.5</b>	<u>NOT ALLOWED</u>
<b>;LI 8;LS 2</b>	four lines per inch

#### PAPER SIZE -- PL Control Word

We just said that ALLWRITE will recalculate the number of lines per page when you change **;LS**. That assumes you are using paper that is 11 inches high. If you're using Legal size paper, which is 14 inches high, or European size paper, which is 12 inches high, or small index cards, you must tell ALLWRITE how big the paper is:

**;PL n default = 66**

That stands for "Page Length", and it is specified in lines per inch, using the current **;LI** value. If you haven't changed **;LI**, then there will be "6" lines to the inch. If you've already set **;LI 8**, and want to specify Legal (14") size paper, you would specify **;PL 112** (14 times 8).

Normal typing paper (8-1/2 by 11 inches) has room for 66 lines per page at six lines per inch, or 88 lines per page at eight lines per inch. However, if you change the number of lines per inch, you won't have to change the number of lines per page, since ALLWRITE will calculate it for you. If you're using legal-size paper (8-1/2 by 14 inches), there are 84 lines per page at six lines per inch, and 112 lines per page at eight lines per inch.

There's one other use for ";PL", and that is when your printer doesn't let you position the paper at the top edge. Some daisywheel printers have this problem: you must start about a half inch (3 lines) down. If you're facing that situation with cut-sheet paper, you can set ";PL 63" to compensate for the lost half inch.

### **JUSTIFIED AND AS-IS PRINTING -- FO Control Word**

This is an important topic. We covered some of it in the tutorial of Chapter 3, but, like most of ALLWRITE's features, it has other capabilities that weren't needed in a beginner's tutorial.

Unless you tell it to do otherwise, the Text Formatter will live up to its name, and everything it prints will be fully formatted. That means the left and right margins will both be smooth (justified), and as many words as possible will be placed on each print line. If a screen line ended with an "ENTER" symbol, but the next screen line contained more text, that "ENTER" symbol will be treated as a blank space, and won't interrupt the joining of text as print lines are filled out. Control word lines can cause text "breaks", as we've already explained, but as soon as more text is found, a new paragraph of fully-justified lines will be formatted and printed.

Most of the time, this mode of operation will be exactly what you want, but there will be times when you won't want screen lines to be joined together for printing. For example, when writing a letter, your address in the upper right hand corner, and the recipient's address on the left hand side, should be printed "as-is", on several separate lines.

When you want the Formatter to switch to "as-is" printing, you can tell it to turn formatting off:

**;FO OFF**

Thereafter, whenever an "ENTER" symbol is found in the text, it will end a print line, and the printed lines will not be right-justified, but only left justified. For example:

```
;fo off
Mr. John Smith
123 Fourth Street
Los Angeles, CA 90001
;sk
Dear Mr. Smith,
;fo on;pp
```

That salutation would be printed just as it was written, but without the control words, of course. However, if the ";fo off" had been omitted, it would have printed like this:

**Mr. John Smith 123 Fourth Street Los Angeles, CA 90001**

The presence of the ;sk ahead of "Dear Mr. Smith" would have still caused a text break, but would not have rescued the rest of the text.

Formatting can be turned back on by:

**;FO ON** or simply: **;FO**

Now, there are other kinds of formatting requirements, and most of them can be controlled by other parameters on this very useful command. Here's what it can do:

<b>;FO OFF</b>	lines are not right justified ENTER symbols cause new print lines.
<b>;FO ON</b>	lines are left and right justified. single ENTER symbols do not cause text breaks. last line of a paragraph is not right-justified.
<b>;FO</b>	"ON" is the default when nothing is specified.
<b>;FO RIGHT</b>	left margin ragged, right margin smooth. single ENTER symbols do not cause text breaks. cancelled by ";FO ON"
<b>;FO ALL</b>	similar to ";FO ON", but the last line of a paragraph is fully justified. useful in making price lists

As you might expect, that list was done with ";FO OFF", and we lined things up with tab stops.

#### Text Justification -- JU Control Word

The ";FO" control word affected two things: the justification of margins and the treatment of the "ENTER" symbol. Sometimes, you may want lines of text to be joined together, but do not want smooth right margins. A ragged right margin is what a typewriter produces, and letters often seem warmer and more personal when they don't scream "I was written by a machine!"

If you just want to control justification (ragged margins), but not concatenation of text, you can use this control word:

**;JU OFF**

It works almost like ";FO", but doesn't affect the treatment of the "ENTER" symbol. In fact, all the parameters of ";FO" can be used with ";JU", and will produce the same results.

#### Suggestion

When printing a letter in "monospace", use "JU OFF" to make it look more personal.

#### CONNECTING LINES TOGETHER -- CO Control Word

If you want to leave text justification alone, but tell the Text Formatter to "break" on each "ENTER" symbol, you can use this control word:

**;CO OFF or ON**

"OFF" tells the Formatter to break on "ENTER"'s, and "ON" tells it to treat single "ENTER" symbols as spaces. You won't have much use for this control word, since ";FO" is more useful most of the time. Note that "ON" and "OFF" are the only acceptable parameters, and that "ON" is the default. "CO" stands for "connect", and the technical word is "concatenation".

**TEXT BREAKS -- BR Control Word**

If you have just two or three lines to be printed "as-is", and don't want to turn formatting off and back on, nor leave blank lines between the printed ones, you can separate them with this control word:

**;BR**

It doesn't use any parameters. Many other control words cause text breaks, including ";FO", ";JU", ";SK", ";PP", and ";CO", and if you've just used another control word that causes a break, you don't need to use ";BR" at the same time.

**PAGINATION -- PA Control Word**

This is another important topic. By default, ALLWRITE will leave one-inch top and bottom margins, print as close to the bottom margin as it can without leaving a "widow" (the first or last line of a paragraph, all by itself), and print the page numbers in the upper right-hand corner of the top margin, beginning on the second page. If you think you might ever want to change any of those things, this is where we'll tell you how to do it.

If you want to start a new page at a certain place in your text, use this control word:

**;PA**

When ";PA" occurs, it will cause a text break. All text up to that point will be printed, the current page will be ejected, and a new page will be started.

The ";PA" control word can do a lot more than that:

<b>;PA</b>	with no parameters, starts new page and adds "1" to previous page number
<b>;PA n</b>	"n" is the number to use on the new page: it resets the internal page pointer
<b>;PA +n</b>	adds "n" to page counter and starts a new page with the new number
<b>;PA -n</b>	subtracts "n" from the page counter and starts a new page with the new number
<b>;PA n,D</b>	"D" stands for "Delay". A new page number is set, but not used until a new page has to be started for other reasons. "n" may be signed.
<b>;PA ODD</b>	forces the new page to have an odd page number. A blank page may be skipped
<b>;PA EVEN</b>	forces the new page to have an even page number. A blank page may be skipped.

We used ";PA ODD" just before every chapter to ensure they would begin on right-hand pages. When we printed the sample letter in Chapter 3, we used ";PA EVEN" to ensure the original text and printed result would be on facing pages. If you know you will need to leave some blank pages for figure insertions, you can use ";PA +n", and if you want to put the "D" after it, you can keep typing to fill out the present page, and know that the page numbering sequence will leave the room you need.

If the paper is still at the very top of an odd-numbered page when "ODD" is used (or an even-numbered page with "EVEN"), the control word is ignored. This means you can use ";PG ODD" at the start of each chapter, then print the chapters separately without having two blank pages ejected unnecessarily.

**WIDOWS AND ORPHANS --WD Control Word**

Believe me, I didn't make up those words. They are the terms used to describe what happens when the first line of a paragraph is the last printed line on a page, or when the last line of a paragraph doesn't fit on the page, and becomes the first printed line, all by itself, on the next page. Isolated lines of that sort don't look very good and should be avoided if possible. The Text Formatter has an effective, automatic "anti-widowing" mechanism, and it works by default.

The drawback to anti-widowing is that, in rare cases, you may need to fit as much as possible on a page, or be required to have 53 printed lines (or whatever) on every page, regardless of how bad it looks. If you're faced with that situation, you can just turn anti-widowing off:

**;WD OFF**

Although there probably won't be any use for it, you can also turn it back on afterwards.

#### **AVOIDING SPLIT TEXT -- CP Control Word**

Anti-widowing only makes sure a single line won't be printed by itself. However, you frequently will want to make sure that several lines stay together. For instance, the underlined topic headings throughout this book had to be kept with the paragraphs that followed them: it wouldn't have looked very good to have a heading as the last thing on a page.

Now, anti-widowing can't do a thing about that, because it isn't possible for a computer program to figure out what your intentions might be in an ambiguous situation. So, when you want to keep several lines together, you must tell ALLWRITE about it. And, the way to do so is with the "conditional page" control word:

**;CP n**

Where "n" is the minimum number of lines that must be available if the next text is to be printed on the current page. If there isn't enough room left, "end of page" will occur, footnotes and bottom titles, if any, will be printed, and a new page will be started. That will keep the necessary lines of text together.

That sort of grouping had to be done hundreds of times throughout this book, so we used Soft Key 'X' for the purpose. It was set to this:

**;cp15;sk2**

When "15" was inappropriate, we just changed it after using the soft key. The ";sk2" was used because we wanted lots of "white space" to make the book more readable.

#### **TURNING OFF PAGE NUMBERING -- PN Control Word**

If you don't want the page number to be printed, but do want ALLWRITE to keep counting the numbers for later use:

**;PN OFF**

To resume page numbering:

**;PN ON**

If page numbering had been turned off during the third page, and turned back on during the fifth page, then pages 4 and 5 would not be numbered, and the sixth page would be numbered "6". If you want the internal counter to stop at the same time printed numbers are stopped, use this:

**;PN OFFNO**

Once that's done, page breaks still will occur, but pages won't be numbered and the internal page counter won't change. If that had been done in the previous example, the sixth page would have been numbered "4".

**PRINTING SELECTED PAGES -- PG Control Word**

Normally, ALLWRITE starts printing with the first text in the file, calls the first page "1", and keeps printing until it reaches the end of the last file in the "append" chain or until you cancel printing by pressing "BREAK". However, if you want to have a large document that uses several files, you might want to print only the contents of the seventh file, which normally begins on page 75, and not process the first six files to do it. That can be done with the ";PA" control word, which was covered earlier.

There also will be times when you just want to reprint a couple of pages in the middle of your document because there was a paper or ribbon jam, or because you've corrected a typo on the page and want an updated copy. In those cases, ALLWRITE has to be able to figure out what text should go on the page(s) to be reprinted, but it doesn't have to take the time to actually reprint any of the preceding pages.

When you just want to print certain pages, the control word to use is shown below. It normally is used only as a "run-time" option, and almost never is included in a file:

**`;PG s [,e]`**

"s" is the starting page number, and "e" is the ending page number. ALLWRITE must begin at the very start of the document, and will process, but not print, pages whose numbers are lower than "s". It will print page "s", if it gets that high, and will keep printing until it has printed page "e". Next, it will prompt you to enter another page range (higher than the first). If you've printed the last, or only, set of pages, just press "ENTER" to indicate that you're done.

To print a single page, "s" and "e" should be the same:

**`;PG 13,13`**

would print just the 13th page. Depending on the speed of your computer and disk drives, and the size and complexity of each page. it might take ALLWRITE a minute or so to process enough text to get through the first 12 pages and reach the 13th one; when skipping pages like this, it takes between three and ten seconds per page.

To reprint the first three pages:

**`;PG 1,3`**

To start printing with the ninth page, and print the rest of the document:

**`;PG 9`**

If the starting page number exceeds the size of the document, nothing will be printed, but it isn't an error.

You can use "PG 9999" to get a quick count on a large document: on a Model 4, ALLWRITE formatted this entire book in 30 minutes (since it didn't have to print anything). ";SM" (summary mode) is another way to do the same thing.

To avoid running out of paper or ribbon during unattended operation, you can use ";PG" to print 100 pages at a time, then change paper/ribbon if necessary, and print the next hundred pages. Each time one range ends, you'll be asked for the next range, so you print your entire document in one, controlled pass.

If you want to re-print page 207 of a book, you could use ";PG", but it might take a half-hour or more to get there. However, there's a better way to do it, and we'll show it to you after explaining one more control word.

**STARTING PAGE NUMBER -- PA Control Word**

The other situation we mentioned was the need to start printing from the beginning of a selected file, but to use a page number other than "1" to correspond with what gets printed from the start of that file. That's done with a control word we covered earlier, and in this form, it's normally used as a run-time option:

**`;PA n`**

"n" is the number to be placed on the first page, and it replaces the default of "1". This is useful when printing a document in sections.

Up above, we pointed out that having to reprint page "207" using ";PG" might take awhile, and that there's a faster way to do it. Since a document of that size consists of many appended files, all you have to do is to start with the file containing page "207", or perhaps with the file just before it. Determine where a new page corresponds with the start of a file, start printing from that file (not the very start of the big document), and specify both ";PA" and ";PG"; SET ";PA" to the page number that appears on the text that happens to be at the beginning of the file you're starting with, and ";PG" to the page(s) you want to re-print. Then, it will take only a minute or less to get to the needed page.

Example:      **`;PA 205;PG 207,207`**

"205" is the number to be assigned to the first page of the file being printed, and "207" is the page to be printed from that file. In effect, the third page of the file will be printed, but it will be numbered "207".

**CUT SHEETS -- CS Control Word**

The final aspect of pagination deals with the use of typing paper, letterhead, or other forms of "cut" sheets: pages that are not continuous forms that a tractor or pin feed can handle. When dealing with cut sheets, it's necessary to tell ALLWRITE to stop at the end of each page to let you insert a new sheet of paper. If your printer only works that way, you probably selected "cut sheets" during installation. However, if you do your rough draft printing with continuous forms, and only your final copies on cut sheets, you normally will want to run in non-stop mode, and tell the Text Formatter when cut sheet operation is to be used:

**`;CS ON or OFF`**

"ON" is the default if no parameter is specified. If the control word is omitted entirely, the default is whatever you selected during installation. If you have more than one printer, the installation values can be different for each printer. This control word usually is entered as a run-time option.

**PAGE TITLES**

This section will show you how to put titles at the top and bottom of each page: the kind you've been seeing throughout this book. Titles can be very simple, that is, just a one line heading; or very sophisticated: up to 45 different titles can be active at any one time, if you want to go overboard.

The simplest title is called a "heading":

**`;HE your words go here`**

Your words will be placed left-justified in the top margin of every page that begins after the ";HE". In addition, the page number will be printed, right-justified, on the same line, unless you've turned off page numbering (";PN OFF"). If any other "top titles" (see below) were used earlier, they will be cancelled.

If anything has printed on the current page, a new heading or top title will not be used until the start of the next page. So, if you want to change the heading or title mid-way in a document, do so just before you force a new page. That is, the ";HE", or any new title, must be placed above the ";CP" or ";PA" control words.

### **Top Titles -- TT Control Word**

A top title is similar to a heading, but gives you much more control over placement. Each title on this page is divided into three parts: left, centered, and right. Also, we are using both top and bottom titles, and the titles on the left hand (even) pages are slightly different from the ones on the right hand (odd) pages. That means you can use four different kinds of titles at once, if you need them. And, there can be nine titles in each of the four areas, for a total of 36 at a time. We don't really expect anyone to use that many, but they're available if needed. Before explaining how they are specified, we'll just mention that a fifth kind of title is used for footnotes, and is explained under that topic.

There are a number of things you can do within titles, and we'll explain all of them before giving examples. Everything described for top titles will also apply to the other kinds of titles.

Let's begin with the most common kind of title: the one that goes on the top of each page. It's called a "Top Title", and is specified like this:

**`;TT /left/center/right/`**

The slashes "/" are required "delimiters". They separate the three sections of the title, and you can use any symbol you like as a delimiter, as long as you use the same one throughout the title, and as long as the symbol doesn't occur in the text that you specify for the title itself. In the next title, you can use a different delimiter.

"left" is anything you like. It will be left justified on the title line. "center" is also anything you like, and will be centered on the page, based on the current left margin and the line length. It will not be centered between the 'left' and 'right' portions, since doing so would put it off-balance for the body of the text. "right", of course, is whatever you want right-justified in the title.

Any of these three sections can be omitted if you don't want any printing in the appropriate area. In fact, any two, or all three can be omitted. If you just wanted something to be centered, you would do it this way:

**`;TT //13TH CENTURY EUROPEAN HISTORY//`**

Note that all four delimiters were used, but there was nothing in the left or right areas.

If you want more than one top title, place the title line number before the first delimiter. It can be any number from 1 to 9, and "1" is assumed if you don't supply a number. By the way, this means you cannot use a number as a delimiter, but you should avoid numbers and letters as delimiters, anyway. When specifying multiple titles, do so in numerically ascending order, since each one will cancel all titles with higher numbers. That's how you can get rid of titles when you no longer want them.

### **Page Numbering In Titles -- \$, PS Control Word**

Automatic printing of the page number is turned off when you define any title. To include a page number in a title, put a dollar sign "\$" in its definition, right where you want the number printed, and ALLWRITE will replace the "\$" by the page number. If you need to use a dollar sign in the title, change the symbol by using this control word:

**`;PS c 'c' is any character not in the title`**

**Where a Top Title Prints**

All top titles print in the top margin, above the body of the paper. By default, one blank line is left between the titles and the body, so if you're using the default values for the top margin (6,1), the title will print on the fifth line of each page, the sixth line will be blank, and the text will start or continue on the seventh line. If you specify more than one top title, the last of them will print on the fifth line, with the others above it, and the first top title on top. If you use more than five top titles, you must increase the value of the top margin. If you want more than one blank line between the titles and the body, you can specify what you want in the ";TM" command:

**;TM 10,2**

would leave room for eight top titles, and there would be two blank lines between the last of them and the body. If you had fewer than eight titles, there would be some blank lines after the last top title.

**Format of Titles**

The formatting specifications in effect when a title is defined will apply to that title, even if formatting specifications change thereafter. The features that matter in this are left margin, line length, pitch, boldface/darkness level, and print wheel (translation).

**Emphasis Marks in titles**

Emphasis marks can be used in a title, and they will affect only what's in the title, not what's in other titles or the body of the document. Similarly, emphasis marks from the body won't affect the contents of a title.

**Canceling Titles**

To cancel a title, replace it by another title, by a "null" title (one that has just four slashes in a row), or by using a title with a lower number: all titles with higher numbers in the same category (top, bottom, odd top, etc.) will be cancelled when any title is specified, so you won't have to cancel them all individually.

If you start using titles, and want to restore the original default we used before any titles were specified, use this:

**;tt //\$/**

**Examples of Top Titles**

Here are several examples of top titles. The things they illustrate apply to the other kinds of titles, also.

<b>;tt /CHAPTER 5/-\$/TITLES/</b>	similar to what we used for this page
<b>;tt//Page \$/</b>	page number and word "page" on right side
<b>;tt 2/Val Jones/Term Paper\$/</b>	second title line
<b>;tt //-\$//</b>	centered page number, surrounded by dashes
<b>;tt /Val Jones/@*2History of Europe@*0/Page \$/</b>	center portion will be boldface

Val Jones **History of Europe** Page 3

**;tt //\$/** restore default numbering method

```
;tt 1//LEARNING ALLWRITE/
;tt 2//Text Formatting/
;tt 3//Titles/
;tt 4//Page $/
```

The last group would result in this kind of printing in the upper right-hand corner of each page:

```
LEARNING ALLWRITE
Text Formatting
Titles
Page 211
```

That looks impressive, but it uses a lot of extra paper in a large book.

### **Bottom Titles -- BT**

Bottom titles print in the bottom margin. The first, or only, bottom title is the first one printed, so if you have three bottom titles, the third one will be closest to the bottom edge of the paper. (if you're accustomed to using IBM's SCRIPT or DCF, they number up from the bottom, which is different from our method.)

Other than printing at the bottom of the page, everything we said about top titles applies to bottom titles. The two groups are entirely independent of each other, and you can use either, neither, or both, as you wish.

### **Odd and Even Page Titles -- OT, ET, OB, EB**

";TT" and ";BT" print your titles on every subsequent page. If you want different titles on the even (left) and odd (right) pages, as we have done in this book, you can specify them separately, using these control words:

<b>;OT</b>	<b>odd page top title</b>
<b>;ET</b>	<b>even page top title</b>
<b>;OB</b>	<b>odd page bottom title</b>
<b>;EB</b>	<b>even page bottom title</b>

Everything we said about top titles applies to each of these. If you specify a top title and then an Odd page top title, the top title will still appear on even pages. If you specify a top title after an even/odd title, you'll replace both of the earlier ones.

In preparing this book, we defined odd and even bottom titles in an "imbed" file, and never changed them thereafter. Each file in the "append" chain imbedded this standard file, which let us preview and print individual sections to check our akurcie [sic]. We set odd and even top titles for each new topic, to make it easier to spot what you, the reader, might be trying to find; and we placed the topic names on the outsides of the page pairs. So, there were four titles in effect at all times as this book was printed.

### **PITCH (FONT) SELECTION -- PI Control Word**

"Pitch" refers to the number of characters printed per inch. Typewriters usually print 10 characters per inch (cpi), and use a character called "PICA". Another popular size is 12 cpi, and it's called "ELITE". Dot-matrix printers have a "condensed" 16 pitch (it's often 17.1, even though it's called 16), and some Daisywheels have 15-pitch wheels or thimbles available (use ";LI 8" with these tiny daisy characters). Almost all computer printers use 10-pitch by default. Most dot-matrix printers also can print narrower characters, in the range of 16-17 cpi, while most daisywheel printers also can print 12 cpi. Some dot-matrix printers can print 12 or 15 cpi as well.

Besides these common pitches, most dot matrix printers can print their characters in double-width, so a printer having 10, 12 and 16 cpi really has six pitches available, any two of which (single and double width) can be used at a time. Some of these printers can intermix single-width pitches on the same line, and others cannot: ALLWRITE supports this intermixing to the extent that your printer allows it. Double-width printing is controlled by Emphasis marks, as listed in Chapter 3 and at the beginning of this chapter.

Some printers treat "correspondence quality" as a different pitch from "rough draft" characters of the same size. When using this higher quality, these printers run slower than they do in rough draft mode. ALLWRITE supports both levels of quality by treating them as different pitches. This applies to printers such as the Microline 92 and Radio Shack DMP series, but not to printers like Epson's, where "Emphasized" printing is just a variant of 10-pitch.

The most interesting pitch of all is "proportional", in which the sizes of the characters vary just as they do in a normal book or magazine. The capital "M" is a much wider character than the lower-case "i", and we picked the two extremes intentionally, to highlight the contrast: "Mimi" shows what we mean. Not all printers have proportional printing capability, and a few that claim to have it really do not, as noted in Chapter 6.

ALLWRITE can use the proportional pitch capabilities of most printers to produce true, right-justified evenly spaced print lines, just as you are used to seeing in a book. In fact, the real beauty of proportional printing is that the reader isn't aware that it's being used, because it just looks right.

To produce smooth left and right margins (full justification), it's usually necessary to add some blanks, or "pads", to each print line. If monospace printing (10, 12, or 16 pitch) is being done, the blanks have to be full sized space characters, and they are placed between some of the words on the line. To make this less conspicuous, the pads are scattered evenly, so as to avoid a rippling or left/right unbalanced effect. ALLWRITE goes one step further: when the pads can't be distributed evenly, some of the extras are placed after punctuation to correspond with the natural pauses between ideas.

Now, if proportional printing is in use, the pads don't have to be full-sized spaces; they can be "micro-spaces", or "dot-spaces". And, while "space" characters are 1/10 of an inch in size, micro-spaces are between 1/60 and 1/200 of an inch, depending on the printer. That makes them a lot harder to see, especially since we can "hide" them between individual letters, not just between the words. One of the reasons ALLWRITE's printing on a dot matrix printer looks so good is that we can allow a little more space between the letters than the printer would if left to itself: The result is less cramped and easier to read.

When using proportional printing, you can specify whether "padding" should be done between the letters (which is the default) or only between the words; and what the minimum spacing between letters should be. You don't have to worry about this sort of thing, since ALLWRITE takes care of it by default, but we will show you how to control it, if you want to do so. Please note that this degree micro-spacing control doesn't work on certain printers, even when they have proportional print capability.

If you don't select a pitch, ALLWRITE will use proportional spacing on most dot matrix printers having the capability, or 10-pitch (PICA) on those that don't. On most daisywheel printers, the default is 10 pitch, since proportional spaced printing won't look good unless a proportional wheel or thimble is used. Later in this section, we will summarize the print wheels and thimbles that are available for some of these printers.

### The PITCH Control Word -- PI

You can select and change pitch by using this control word:

**;PI n "n" = 0, 1, 2, 8, 9, 10, 11, 12, 13, 16, 17, 20, 21**

"n" is the pitch to be used, and zero "0" selects proportional spaced printing. If your printer cannot use the pitch you select, the next lower-numbered pitch will be used. For example, the Epson MX-80 does not have 12 pitch, but the FX-80 does.

Although ALLWRITE recognizes ten numeric pitches, the four common ones are:

### **0 (proportional), 10, 12, and 16**

Daisywheel printers use "15" pitch instead of "16", but you can specify "16" for compatibility with your dot-matrix printer, and ALLWRITE will use "15" automatically.

As soon as the ";PI" control word is encountered, ALLWRITE will switch to the new pitch, even in mid word, if your printer allows it. Titles will continue to be printed in whatever pitch had been in effect when those titles were first defined, even though the body of text on the surrounding pages may be in a new pitch.

If you just use ";PI" without a number, the original default pitch will be used thereafter. Here are some examples of pitch selection:

<b>;PI 16</b>	selects 16 pitch on dot matrix
<b>;PI 10</b>	selects 10 pitch on most printers
<b>;PI 0</b>	selects proportional spacing This doesn't work on all printers! Diablo widths are for "THEME 11" wheel Spinwriter widths are for "BOLD PS" thimble
<b>;PI 12</b>	selects 12 pitch on some printers This doesn't work on all printers selects high quality Elite This only works on a few printers

#### **Microline and DMP printers**

If your printer has a "correspondence quality" version of certain "monospace" pitches, you can choose between fast printing for rough drafts and better quality for the final copy. To do so, specify a pitch number that is "1" higher than normal. For example, if you're using 10-pitch for rough draft, then 11-pitch will select correspondence quality; the higher-quality version of 12-pitch is 13-pitch, etc.

#### **Epson Printers**

Correspondence quality printing is selected in different ways on different printers. For example, Epsons can switch print quality only by using boldface ("EMPHASIZED"), and they can do that only in 10-pitch. Attempts to use Emphasized printing in 12 or 16-pitch will not work on these printers, although Double-strike (boldface level 2) often will work in any pitch. If boldface does not work in certain pitches on your printer, check your printer manual to see whether it's a hardware limitation.

#### **Other Printers**

If you're using a daisywheel printer, you will also need to read about "Proportional Printing on Daisywheels," which is the topic after the next one. We've treated it separately in order to put all the information you'll need in one place. If you plan to use proportional printing on dot matrix or daisywheel printers, you will need to read the next topic, ";SD".

**Downloadable Printers**

Some dot matrix printers have a feature called "downline loadable fonts". We used that for some of the headings in this book. If you have a suitable printer, please see the ";DF" (Download Font) control word later in this chapter: it is used in connection with ";PI". Please note that we only plan to support a few of the printers having this capability.

**CONTROLLING MICRO-SPACE PADDING -- SD Control Word**

The name of this topic has a forbidding "technical" ring to it, but that really isn't the case. As we said earlier, "padding" is the method used to produce smooth right margins, and if you're using a proportional spaced character set, ALLRIGHT offers you some control over how this is done. If you like the way your documents look with the default values, you don't need to use this information, but there are times when it might be useful.

Micro-space padding is defined by:

**;SD n [ ,NO or YES ]**

";SD" stands for "spacing definition." 'n' is the minimum number of micro-spaces to be placed between characters when proportional printing is in effect. A micro-space is the smallest horizontal spacing unit your printer allows, and varies from 1/60 of an inch to 1/200 of an inch (printers planned for the future may have even better resolution than this). The minimum value is added to what's going to be on each print line, regardless of how much padding will also be needed. Dot-matrix printers leave only one micro-space ("dot-space") between letters, and that generally looks too crowded, so ALLWRITE can let you force some extra micro-spaces between the letters to produce a better appearance. Daisywheel printers use good spacing to begin with, so we didn't have to make any adjustments on them. If you're using an EPSON in proportional pitch, ";SD 0,Y" is the only setting that will work.

";SD" also controls "padding" for right justification. By default, the pads are placed between the letters as well as between the words, but if you want all the padding (other than the minimum you specify) to be placed between the words, and none of it between the letters, specify "YES" after the "minimum" number. This book was printed using inter-character padding; many newspapers use that method also, and books are done both ways. Books also use something called "kerning", whereby a lower case "i" can be tucked under the overhang of an "f". ALLWRITE doesn't do that, but "DOTWRITER" does, using dot-matrix graphics.

Here are some examples. If you are using a dot-matrix printer and want to increase the minimum spacing between letters:

**;SD 3**

Here is a line printed several times with micro-spacing values of 0, 1, 3, and 5:

- (0) Fourscore and seven years ago
- (1) Fourscore and seven years ago
- (3) Fourscore and seven years ago
- (5) Fourscore and seven years ago

If you want to fit as much as possible on each page, or think our defaults are too large, you can make the printer work the way it does normally by doing this:

**;SD 0**

No extra padding will be done then, but pads needed for right justification still will be added as necessary.

These two paragraphs show the difference between inter-character and inter-word padding. The first paragraph was done normally, using inter-character spacing, while the second one used ";SD 0,Y" to force inter-word spacing only. As you can see, the same words are on each line, but the appearance is subtly different.

These two paragraphs show the difference between inter-character and inter-word padding. The first paragraph was printed normally, using inter-character spacing, while the second one used ";SD 0,Y" to force inter-word spacing only. As you can see, the same words are on each line but the appearance is subtly different.

### **PROPORTIONAL PRINTING ON DAISYWHEELS**

This section applies only to Spinwriters and Diablo-compatible daisywheel printers, not to Radio Shack daisywheels. With these, you can use several different proportionally-spaced print thimbles and wheels. To do so, you must also control pitch and possibly character sequences, because that's how those wheels are constructed. This section will explain how pitch (width) and translation (sequence) can be specified, but if you just want to use the examples below, and skip the explanation, it's O.K. with us.

When ";WT" is used in the examples below, it must be at the start of the file; the ";PI" that uses the defined tables should appear normally when you want to select that pitch.

#### **DIABLO and DIABLO-compatible**

10-pitch wheel	:pi 10
12-pitch wheel	:pi 12
Theme 11 wheel	:pi 0
Theme 10 wheel	:pi 1
any Qume WPS wheel (ESSAY, MODERN, TITLE)	:pi2;tr0
ORATOR wheel, monospace four lines per inch	;wt M, orator/tab :pi M ;ls 1.5
ORATOR wheel, proportional four lines per inch	;wt P, oratorpr/tab :pi P;ls 1.5
Gothic 15 wheel, proportional looks great at 8 lines/inch	;wt G,prop15/tab ;li8;pi G

#### **SPINWRITER**

10-pitch wheel	:pi 10
12-pitch wheel	:pi 12
BOLD PS	;tr @;pi 0
EMPORER PS	;tr @;pi 1
ORATOR monospace four lines per inch	;wt M,orator/tab :pi M ;ls 1.5
ORATOR proportional four lines per inch	;wt P,oratorpr/tab :pi P,ls 1.5

If you use a translated sequence (;TR @") and switch back to a normal wheel or thimble, you should also switch back to the non-translated sequence by specifying ";TR" with no parameters.

**Character Widths**

The character sizes on each print wheel or thimble are stored by ALLWRITE in "width tables", some of which are contained in the printer tables you must keep on your working disk, and some of which are in separate tables on the ALLWRITE distribution disk. Although dot matrix printers can switch pitches by themselves, daisywheel printers need some help: we can alter the character spacing, but you must change the wheel or thimble. If you don't, the results won't look very good, because characters will be unevenly spaced or print on top of one another. That's fine for rough drafts, but not very good for finals.

The ";PI" control word can accept other values besides the ones shown above, but only on a printer-dependant basis. The information below doesn't apply to dot matrix printers or to the Radio Shack daisywheels available as of the beginning of 1984. There is more than one proportional width table built into the Diablo and Spinwriter printer tables, and you can select them by using the pitch numbers "1" (Diablo and Spinwriter) or "2" (Diablo only).

Note to Spinwriter owners: There are two kinds of Spinwriters. The ones whose numbers end in zero (such as 3530, 5510, 5520, 5530, 7730) are "native" Spinwriters, and whatever we say below about the Spinwriter applies to those printers. The ones whose numbers end in "5" (such as 5515, 5525, 7715) are "Diablo compatible", so most of the printer controls created during our installation procedure match Diablo requirements. However, the translate and width tables in the printer table match normal Spinwriter requirements, since your printer uses the same thimbles as do the "native" models.

Note to Diablo-compatible owners: Not all print wheels will work on your printer. The wheels with 96 spokes do work, but there also are wheels with 92 or 88 spokes, and there are metal as well as plastic wheels. Many printers can use only the 96 spoke plastic wheels. If your printer is among them, wheels with fewer spokes, such as the ones in "LEGAL" sequence, simply cannot be used: the spokes won't line up when the printer rotates them. Also, printers that only accept plastic wheels can break if you try to use metal wheels: the metal ones are too heavy to be spun quickly enough on those machines.

Besides those choices, there are several less-frequently used tables on the distribution disk. You can use them if you copy them to your own disks and use the ";XT" and ";WT" control words described later on. When you do so, you will have to assign a letter of your choice (A-Z) to each wheel/table, and use that letter in the pitch command when necessary.

```
;WT G,PROP15/TAB
;PI G
```

"G" is arbitrary, but if you were using a Gothic 15 wheel, it would make it easier to remember the association. We will explain this further when we cover the use of the disk tables.

**Built-in Daiseywheel Width Tables**

The wheels and thimbles built into our printer tables are:

DIABLO and DIABLO-Compatible:

PITCH 0	- THEME 11	- used to print this book
PITCH 1	- THEME 10	- slightly smaller proportional
PITCH 2	- QUME WPS	- larger proportional, translated

NEC SPINWRITER:

PITCH 0	- BOLD PS	- excellent proportional thimble
PITCH 1	- EMPEROR PS	- slightly smaller proportional

Both NEC thimbles are in translated sequence.

**Disk-Resident Daisywheel tables**

The files listed below contain width tables, but they will be needed only if you're using a special purpose printing element. None of them are "translated" and all can be used with Spinwriters and Diablo-compatible printers:

MONO10/TAB	- simulated proportional with PICA wheel
MONO12/TAB	- simulated proportional with ELITE wheel
PROP15/TAB	- proportional with GOTHIC 15 wheel
ORATORPR/TAB	- simulated proportional for ORATOR wheel
ORATOR/TAB	- monospace for ORATOR wheel
ANELIA/TAB	- proportional for Brother

**NON-STANDARD PRINT WHEELS -- XT Control Word**

There are several considerations: on some thimbles and wheels, the characters are not in their normal positions, and if you try to use these to list something like an accounting report, you will get meaningless printed results. These thimbles and wheels are in different "sequences" than the normal ones, and the printer manuals generally warn you that to use them "may require special software." Well, ALLWRITE has that "special software" built right into it, so you can use those beautiful fonts from now on. To do so, you will need to know about four control words, including ";PI" (pitch), which we just covered. The others are:

:XT	identify a disk-resident translate table
:WT	identify a disk-resident width table
	(XT and WT usually use the same disk table)
:TR c	start using a certain translate table
:PI c	start using a certain width table

If you need to use disk resident-tables, ALL of them must be specified at the beginning of your document file, before ANY printing occurs. If you expect to use these tables most of the time, we recommend you identify them in the "defaults" file for your printer, so that you won't have to remember to type their names into every document you write. The tables are identified through this control word:

**;XT c, fileid**

";XT" stands for "translation table." It identifies a special character sequence that you plan to use later on, but doesn't start using it: you must use ";TR" afterwards for that.

'c' is any letter of your choice, A-Z; 'fileid' is the name of the disk file that is the table you want to use. There can be 26 such tables in effect at once, if you need that many and can find the wheels. All ";XT" control words must appear together, before any ";WT" or ";DF" control words; and these 3 control words must appear before anything else except ";CM" or ";BA". Remember, if you don't need these capabilities, or if the built-in tables meet your needs, you should not use these control words or tables. We're covering something now that only applies to a few people in unusual situations.

If you think this does apply to you, there still may be a loophole: the translation table for the SPINWRITER, and the most common translation table for the DIABLO, are built right into the printer tables, so you won't need to use ";XT" for those: see ";TR", coming up soon.

**Width Tables -- WT Control Word**

We've explained that the most common widths are already built into the printer tables and are directly available as pitches "0", "1", and "2". If you want to use some other width(s), you can specify the table(s) to be used by placing the following control word at the beginning of your document, just after any ";XT" control words:

**;WT c,fileid**

";WT" stands for "width table". It identifies a table that may be used later on, but doesn't start printing in that width: You must use ";PI" afterwards for that.

'c' is any letter, A-Z, and "fileid" is the disk file containing the width table you want to use. There can be up to 26 tables in alphabetical sequence in any one document, if you need that many. Note, that built-in width tables are identified by numbers, while tables read from disk are identified by letters. Also remember that you won't use disk tables for the widths that are built into the printer definitions. The available tables are the ones listed earlier in this section.

### Using a Translation Table

To begin using a non-standard (translated) sequence, use this control word:

**;TR c**

where "c" is one of the letters in an ";XT" defined at the start of the document, or the at-sign "@", which selects the internal translation table.

Most of the time, you won't need to use ";XT", because the special SPINWRITER thimbles all use the same non-standard sequence, and many of the DIABLO-compatible wheels (made by QUME, as it happens) use the "WPS" sequence. If you are using one of those wheels or thimbles, the translation table for it is already built into the printer table, and can be activated at any time by this special form of the translate control word:

**;TR @**

the "@" sign selects the built in table. If you're using a Spinwriter, that table works for the BOLD PS and EMPEROR PS thimbles. If you're using a Diablo-compatible, the built-in table works for all "WPS" sequence wheels, including "ESSAY", "MODERN", and "TITLE".

Note: Other uses of ";TR" will be covered later.

### Using a Width Table

To activate a width table, use the ";PI" (pitch) control word we covered earlier. To use an internal table, specify the pitch as a number: "0" selects the primary proportional width table. To use an external table, previously identified by the ";WT" control word, specify the pitch as a letter: the same letter you used in the ";WT". You can intermix internal and external pitches as necessary, but remember that pitch changes usually mean you'll have to change print wheels unless you use the ";WC NO" control word.

### Wheel Changes -- WC Control Word

On a dot matrix printer, a pitch change is performed automatically by the printer, and you don't have to make any mechanical adjustments. On a daisywheel printer, a pitch change often means you need to change wheels, and a translation change always means you need to change wheels (or thimbles). It's easier to change a wheel when the printer isn't printing, and ALLWRITE gives you two ways to do it: the "PAUSE" emphasis mark, "@>", and the control word shown below:

**;WC YES or NO**

";WC" stands for "wheel change". If you use "YES" and there is a pitch or translation change (;PI or ;TR), then, when the text following the change is about to be printed, the Text Formatter will stop printing and ask you to press ENTER after the wheel has been changed:

**CHANGE WHEEL TO PITCH nn, SEQUENCE x  
PRESS <ENTER> TO CONTINUE**

If you use ";WC NO", printing will just continue, and you won't get that message. The non-stop mode works well for pitch changes with rough drafts, but isn't too feasible if you change the translation sequence. In practice, it may not be so bad, because you probably will use a set of wheels or thimbles that are all in the same sequence, even if they have different pitches.

The "Pause" emphasis mark "@ >" tells ALLWRITE to stop after printing the characters immediately before the pause, and to prompt you to press "ENTER" after making some change. In this case, ALLWRITE doesn't know what you want to do, so if you use this, you'll have to keep track of why the pauses were inserted.

Note to former NEWSCRIPT users: the "PAUSE" method was retained mostly for compatibility with your NEWSCRIPT files, but the ;"WC" method is a lot easier to use, since it's automatic.

**DOWNTIME LOADABLE FONTS -- DF Control Word**

This section applies only to the DMP-2100P and the LQ-1500 on Models 4 and 4P. Some dot matrix printers let you define the shapes of characters they will print by "downtime loading" new, temporary character patterns. ALLWRITE supports this feature on certain printers but not on all of them: as we write this book, only the Radio Shack DMP 2100P and the Epson LQ-1500 are supported. Both are 24 pin printers capable of very high quality printing (we used one of those printers for this book). We do not plan to try to support 8-pin printers, since we have very small fonts.

You need four things (besides the printer) to use this feature with ALLWRITE:

- a downtime loadable font file
- a width table for that font
- the ";DF" control word
- the ";PI" control word

The font file must be in a form that is acceptable to the printer "as-is", with no interpretation or modification needed by the program that transmits it to the printer. It must contain all the printer control codes, as well as the character patterns themselves. If you have patterns that do not meet these requirements, you may be able to use them by loading them into the printer before running ALLWRITE. If you have patterns that meet these requirements, ALLWRITE may be able to load these for you as needed, and switch/reload them to let you use several of them in each printed line. The "extension" portion of the font file names must always be "/FNT", or ALLWRITE won't find them.

Note: Downtime loading takes 8-12 seconds per font, because of the size of the disk files and the limited acceptance rate of the printers.

The width table must be in ALLWRITE standard format (the name format we use for daisywheel printers), and must correspond to the appropriate font image. The width file must give the width of each character in ASCII order, beginning with the "blank" character in the second byte of the sector; '64' must be added to each width, so that they will appear to be upper-case letters: "A" stands for a width of "1". (If that sentence is gibberish to you, just use the fonts and don't worry about how they work; if you are a computer programmer, you can use our daisywheel tables as examples of the format.)

PROSOFT sells "font disks" for the printers whose download feature we support. Please contact us for details if this applies to you.

If you have one or more font files and width tables, it is easy to use them with ALLWRITE. In fact, only two steps are involved:

**define each font file with ";DF"  
activate a font with ";PI"**

The ";DF" (Define Font) control word is identical to the ";WT" control word, and in fact, it must be intermixed alphabetically with any ";WT's" if you use both of them (usually, you will just use one or the other). Like ";WT", ";DF" must appear at the very beginning of the document file, after any ";XT's but before anything else. The format of the control word is:

**;DF c ,fileid**

'c' is a letter from A to Z (upper and lower case are identical) that you will use in the ";PI" (pitch) control word to select the font when you want it used for printing. 'fileid' is the name of the WIDTH TABLE that is associated with the downloadable file. Do not specify the download file!

Example:

Font file name: FLOW3/FNT  
Width table name: FLOW3/TAB

Note that the "/TAB" file, not the "/FNT" file was specified; and that there was a "/FNT" file with the same root name "FLOW3" available. The width table files do not have to be "/TAB", but the downloadable files must be "/FNT", and must have the same root names as their corresponding width tables. There is no flexibility in this: the feature will work only if you follow these naming conventions!

Once you have specified all the ";DF's" you plan to use, you can just write a normal document, using the ";PI" control word to select any combination of hardware and downloadable fonts you need. You can use the fonts in multiple columns, headings, titles, footnotes, etc. To select a font that was defined with ";DF", just use the "letter" form of the pitch control word:

Example:      **;pi F**

will select the "FLOW3" font defined in our earlier example.

### **Notes about Downline Loaded Fonts**

As we explained under the ";PI" control word a few pages ago, a pitch change does not add a blank character for you. That means you can switch pitches in mid-word, and still get correctly aligned margins; but it also means that, if you want to switch pitches between words, you must type the blank yourself. When you do so, the blank must be the first character of the text in the new pitch, and not the last character of the text in the old pitch.

It can take 5-20 seconds to read a downloadable font from disk and transmit it to the printer. This is a function of the disk speed and printer acceptance speed. Fonts stored on a hard disk, or in a "MEMDISK", load faster than fonts on a floppy disk. Use of an external printer buffer can speed things up also.

If the ALLWRITE Text Formatter has to ask you any questions while a downloadable font is being loaded, the download will stop until you finish answering the question and press "ENTER". You will most likely encounter this with the "EJECT PAGE" question that text being printed is not yet at the end of the document.

Some downloadable printers support more than one loadable font at a time. ALLWRITE takes advantage of this by switching from one font buffer to another; it tries to replace the font that has not been used for the longest time ("least recently used" is the technical term).

### **MISCELLANEOUS FORMATTER CONTROL WORDS**

There are several major, specialized topics left and they include some of ALLWRITE's most powerful and flexible features. Before covering them, there are a few little topics that didn't seem to fit anywhere else, so we will cover them here.

**SUPPRESSING SPECIAL FEATURES -- NO Control Word**

If you use a control word, ALLWRITE assumes that control word should be processed. There will be situations in which you won't want certain things to happen, even though there are control words present for them. For example, you may want to preview just one file in the middle of a chain of appended files, without seeing the ones that follow. In that case, you will want ALLWRITE to disregard the ";AP" control word.

You can suppress certain features individually by using the control word shown below, followed by as many of the listed feature abbreviations as you will need suppressed:

**`;NO list,list,list...`**

Example:      **`;no tc,ix`**

Separate the abbreviations by commas. ";NO" usually is used as a run-time option, but if you're using table of contents and/or index references, and want to bypass them in rough drafts, you may want the ";NO" in the document itself, or in a file that is imbedded into every segment of the append/imbed chain. (That's what we did while developing this book.)

These are the things that can be suppressed with ";NO":

<u>ABBREV</u>	<u>SUPPRESS</u>
AP	appends
FN	footnotes
FP	footnotes, but still print their numbers
IM	imbeds
IX	index collection
TC	table of contents collection.

**DARK PRINTING -- DA Control Word**

The normal way to control boldface is through the Emphasis mark, "@ \*n", where 'n' is a number from zero to nine. However, if you want to print an entire document in boldface, or Emphasized mode, as it's called on several good dot matrix printers, you can use this control word instead of, or in addition to, the Emphasis mark:

**`;DA n n=0 to 9`**

The darkness values from the control word and the Emphasis marks in the text will be added together to produce the actual boldface level that will be used at any particular time. The total cannot exceed nine.

This control word is most useful as a run-time option, because your rough drafts generally will be printed at maximum speed, and only your final draft will be printed with maximum quality. For example, when you're ready to print the final copy on an Epson MX-80, you could enter this as a run-time option:

**`;DA 1`**

That will invoke "Emphasized" mode printing for the entire document. By the way, if you use that particular value, it will be effective on an Epson only in 10-pitch: Emphasized mode is disabled in printer hardware at 12 and 16 pitch. If you want to use one of those pitches on that printer, you will have to select Double-strike: ";DA 2" which prints much slower, since the printer makes two passes on every line.

**NOT UNDERLINING BLANKS -- US Control Word**

For most purposes, the best way to do underlining is with the "@ \$" and "@ %" Emphasis marks. If you want to underline words, but not the blanks between them, use this control word, followed by the text that is to be underlined:

**:US your words go on the same line**

";US" stands for "underscore". It doesn't cause a text break, so the words can be in mid-sentence, if that's where they belong, and you can use several of those control words in a row to underline a lot of text.

**your words go on the same line**

**COMMENTS IN TEXT -- CM Control word**

If you want to include some notes or comments in your text files, but don't want those comments to be printed, you can use this control word:

**:CM anything you like can follow it**

It must be the last control word on the "line", since everything after it, until the next "ENTER" symbol, will be treated as part of the comment, and therefore ignored. This control word does not cause a "break" in text formatting, so if ";FO ON" is in effect, the word just before the "ENTER" symbol that preceded ";CM" will be followed by the word just after the "ENTER" symbol that ends the comment.

**ENTERING TEXT FROM KEYBOARD -- KE Control word**

"Keyboard entry" is a way of filling in missing information at print time. It differs from Form Letters, which mostly use information from a disk file, and it differs from Mini-Edit, which mostly is intended for making last-second corrections of minor errors. Anything entered through Keyboard Entry will be fully formatted according to whatever specifications are in effect at the moment. Moreover, you can enter control words and emphasis marks as well as text, so this offers you a flexible way of inserting amounts, reminders, or personal touches to a standard letter.

Keyboard entry is activated by this control word:

**:KE [n]**

where 'n' is the number of lines to be read from the keyboard. A line ends when you press "ENTER", and shouldn't be more than two screen lines in length. If 'n' is zero or omitted, you can type as many lines as you like, and exit keyboard entry by pressing "ENTER" an extra time without having typed anything in response to the most recent prompt.

ALLWRITE will display any text that follows 'n' or ";KE", so you can use this for a prompting message to remind you of what to enter.

One good use for keyboard entry is in customizing an occasional Form Letter in a situation where you don't need the full power of Allwrite's Form Letter capability and don't want to set up a mailing list. A physician's office might use this as a way of typing in a date in a reminder letter, for instance.

Examples:

- :KE 3** read 3 lines from keyboard
- :KE** read one line
- :KE 0** read until just "ENTER" is pressed.

**STOPPING FOR OPERATOR INTERVENTION -- ST Control Word**

When we explained how wheels could be changed on daisywheel printers, we showed a use for the "pause" Emphasis mark: "@ >". It is a way of pausing the computer at the output, or printing stage of text processing. Now, if you want to pause the computer at the input, or disk reading stage, use this control word:

**;ST your message goes here**

This is useful when the next control word is ";AP" and the file to be appended is on another disk. You can pause processing at that point, switch disks, and then press "ENTER" to tell ALLWRITE to continue. The Text Formatter will display the message that follows ";ST", store the appends File name, and then continue after you've pressed "ENTER".

It's O.K. if the next screen line is not an Append, because you might want processing to stop for some other reason we haven't anticipated. By the way, printing won't stop during this pause, unless the Text Formatter runs out of "buffered" material (text it's already processed, but hasn't sent to the printer yet because of the speed differences between the computer and the printer).

**CARBON (MULTIPLE) COPIES -- CC Control Word**

We aren't talking about real carbon paper here, but multiple printed copies of a document, printed one after another. This is different from Form Letters, which is one of the major topics we will cover soon, because names and addresses change in Form Letters, but not in carbon copies.

The control word, which normally is used as a run-time option, is:

**;CC n 'n' is the total number of extra copies to print**

If 'n' is omitted, a total of two copies, including the first one, will be printed. If you hit too many keys, you'll use up a lot of paper, because "65535" is the upper limit on this one, and it doesn't stop for coffee unless you press the "BREAK" key (sorry about that). This control word is ignored when Form Letters (";RD") are being printed.

**Note to NEWSCRIPT Users**

In NEWSCRIPT, the control word ".CC" meant "conditional column", but that specification is handled in ALLWRITE by ";CP" (conditional page, which recognizes that a new column, not a new page, may be sufficient).

**MULTIPLE COLUMN PRINTING -- CB Control Word**

"Columns" can refer to columns of numbers in a table, or to columns of text in a magazine or newspaper. The "table" kind of column is done with tabs, as explained in Chapter 4. Up to sixteen columns of text can be printed by using this control word:

**;CB 2**

All text after that control word would be printed in two side-by-side columns, with a half inch "gutter" between them. A "gutter" is the white space left between the columns. All the columns for a given line on the paper are printed at once, so this works on all printers. Since a full page of text has to be stored before printing can begin, there may be a short delay before the first page starts to print. Thereafter, the Text Formatter will be so far ahead of the printer that no further delays will be noticed.

To return to single column printing:

**;CB 1**

You can change the number of columns on a page as often as you wish. If there are footnote references within some of those column groups and you change the number of columns in mid-page, it may not be possible to print the next column group on the same page, since the footnotes are printed in the same column size as the text that references them. See "footnotes" for further information later in this chapter.

If you want to force printing to start in a new column at any time, use the control word with no parameters. This is similar to forcing a new page, but only moves to the next column:

**;CB**

You are not limited to two columns, nor to half-inch gutters, since the full form of the control word is:

**;CB n [,w] [,B or U]**

'n' is the number of columns to be printed, and can be anything from '1' (normal printing) to '16' (if you have a 15" carriage and are printing in 16 pitch). 'w' is the width of each column in tenths of an inch. If omitted, the column widths are calculated by allowing a half-inch gutter between each column, and dividing the remaining space evenly among the columns. In monospace pitches (10, 12, etc.) the total width of the print lines will be slightly shorter than the specified line length when the column and gutter widths cannot

be divided evenly. You may be able to bypass this by increasing or decreasing the column width by '1'.

"B" and "U" are used to select "Balanced" OR "Unbalanced" column printing. When columns are balanced (which is the default), they will all be the same length (end on the same line of the page), so the last page of a chapter or book will have no printing near the bottom. When columns are unbalanced ("U"), the first column(s) will be full length, while the last column may be much shorter. If the total number of printed lines is not an exact multiple of the number of columns, balancing may be one line off.

Certain conditions can force unbalanced column printing even if "U" was not specified. For example, forcing a new column with a simple ";CB" will override balancing; so can paragraph anti-widowing.

Certain combinations of emphasis marks and control words cannot be used within multiple columns. For example, you cannot change the number of lines per inch within a multiple column area or attempt a half-line space, since doing so would mean that text in one column wouldn't be on exactly the same line as text in the adjacent column. Printers don't work that way, and the only way to get that kind of printing would be to print one column, reverse feed the paper, then print the next column. If your printer doesn't support pitch changes within one line, you should avoid using the ";PI" control word within a multiple column area: the pitch change could fall in the second column, conflicting with the printer's limitations.

Hyphenation is more important in columns than in normal printing, because inter word padding can cause big gaps without it.

It is possible to run out of "memory" in the computer if you define too large a page. An extreme example would be a 13 inch line (";LL130") using 16-pitch at eight lines to the inch, with numerous footnote references: the text alone would contain almost 15,000 characters, and there isn't that much room available. A normal page of proportional text uses about 5,000 characters, and can easily fit in your computer's memory. On a Model I or III, you probably will run out of memory if you try to preview ";VI" multiple columns with small pitches. A partial solution is to re-run ALLWRITE without the "ALK" keyboard driver. That will make several hundred more bytes of space available to the Formatter, and may let you get past the limitation.

**ERROR CONTROL -- ER Control Word**

It would be nice if we could tell you that this feature lets you control errors so that they won't happen to you. If we knew how to do that, we wouldn't have to write computer programs for a living. Instead, we will settle for letting you tell the Text Formatter what to do when a control word error occurs.

Possible control word errors are listed and explained in Chapter 8, and include such things as "unknown control word" and "invalid value" (a top margin that's bigger than the page length, for instance). When such errors occur, the Text Formatter will switch to "Summary" mode immediately, and only print error messages and the first line of each page, unless you tell it to do something else:

**;**ER** Quit or Summary or Edit**

"Q" tells the Formatter to stop processing immediately and display the "Exit menu" (what do you want to do next?). "S" (Summary mode) is the default, but "E" will activate "Mini-Edit," and let you try to correct an error from the keyboard so that printing can continue. If you want to use "E", you must do so before any errors occur.

";ER" is useful both as a run-time option and in your normal defaults file.

**Printing Imbedded File Names -- NA Control Word**

This feature is useful when you are proof-reading rough drafts of large documents. It prints the name of each imbedded and/or appended file as that is encountered, and makes it easier for you to identify which file needs to be edited as you make revisions. Each file name is printed on a line of its own, thereby changing the pagination slightly; please take that into account if you decide to use the feature:

**NA ON or OFF or A, I, R**

";NA" alone defaults to "ON", and causes the file names to be printed when an "append" or "imbed" is started, and also when an "imbed" ends and control returns to the "higher" file. "OFF" disables the feature.

If you only want some kinds of names to be printed, you can use "A" (appends only), "I" (imbeds only), or "R" (returns from imbeds only). You can also use two or three of these together. If you chose to be selective, you must not use "ON" at the same time. For example, If you just want to trace the start and the end of each imbed:

**;**NA** I,R**

We have now covered all the normal word processing features you are likely to need for letter writing and most other everyday purposes. If you've just been reading along out of interest (or more likely, insomnia), this would be a good time to take a break. The next topic is Form Letters, and it should be read when you are feeling alert, or you won't believe a small micro-computer can do what we are going to show you.

**FORM LETTERS**

In a "Form Letter", the names and addresses change from copy to copy, but just about everything else stays the same. Form Letters are used in two different ways: 1) as a prototype letter that can be modified each time a copy of it has to be sent out, and 2) to send a lot of letters all at once. In this section, we will explain how you can use ALLWRITE for both of these purposes. If you are a former NEWSCRIPT user, your existing Form Letters and Mailing Lists still work with ALLWRITE.

A "prototype" letter is easy to use because it's all done with facilities of the Editor. First, you can write several standard letters and store them on disk. When you need to send one of them, you can "Edit" it, assign a temporary file name in order to preserve the original copy, and then type in the name, address, and other specific information that

will tailor it to the current use. After the tailored copy has been printed, the temporary copy can be "killed" if it's no longer needed, or archived for future reference. The ";KE" control word is quite effective with these low-volume letters.

There's an on-going need for small-scale personalized prototype letters, but it isn't the main purpose of the present topic. What we really want to show you is how to merge a mailing list with a prototype letter so that dozens, hundreds, or even thousands of "personalized" letters can be printed one after another, automatically. ALLWRITE has excellent, easy-to-use facilities for doing this, and the only real limit will be the time it takes your printer to make each copy: a thousand letters at two minutes per page takes about 33 hours to print, and that really isn't too good for most micro-printers.

To produce Form Letters with ALLWRITE, you need a prototype letter and a mailing list. Each one must be in certain formats, which we will explain here. The letter will always be in a disk file, and the mailing list usually will be in a disk file, although names and addresses can be typed in from the keyboard. Usually, you will want the list to be in a disk file, since it can be re-used easily for other letters. We will show you different ways to do this, and give examples of each method. We also will show you how short, high volume letters can be printed efficiently.

### **SELECTING FORM LETTER CAPABILITY -- RD Control Word**

The prototype letter will be a normal letter, but it will contain control words and "variables" to indicate where a name, address, title, amount, or other information should be inserted from the mailing list. A "variable" is just a symbol that stands for something else. For example, when we showed you how to set the line length, we used the expression, ";LL n", where 'n' stood for a number that you had to supply. 'n' was a 'variable' there, since it had to be replaced by a number.

Some approaches to Form Letters are simple but limited; others take a little more effort to prepare, but can produce letters so personalized that no one would believe they were generated by a computer program (particularly if you don't right justify them). Most of these ways use this control word:

**`;RD c,fileid`**

#### Examples:

<b><code>;RD 3</code></b>	read 3 lines from keyboard
<b><code>;RD 4,XMAS/LST</code></b>	read 4 lines per letter from disk
<b><code>;RD @ CUST/NAM</code></b>	read a flexible number of lines from disk -- one N&A set per letter

"N&A" is an abbreviation for "Name and Address". There are other, fancier forms of ";RD", and they will be covered later on. The simple form shown here will serve most of your needs.

Examining the examples is the easiest way to understand how to prepare a Form Letter, but we will explain what the notation means first: 'c' may be a number or a symbol such as "@", and the optional 'fileid' identifies a mailing list on disk. If 'fileid' is omitted, you will be prompted to type each name and address from the keyboard, a method we don't recommend unless you only expect to do one or two of these letters at a time. The ";RD" control word has some other capabilities, but we won't discuss them yet.

If 'c' is a number, the Text Formatter will read that number of lines from the keyboard or from 'fileid' and then print them as-is in the next lines of the document. By repeating this cycle, it will print one Form Letter for each set of names and addresses selected from the list.

If 'c' is a character, usually the at-sign "@", the Text Formatter will read the next set of lines from the disk file and include in it the letter. A new letter will be printed for each set of lines, so a 'set' should contain all the necessary information for one person on your list. Each of these sets can have a different number of lines, so this method is more flexible than the first one, since it allows for the presence or absence of company names, mail codes, etc. We will show this in two of the examples later on.

ALLWRITE can create Form Letters in 4 different ways:

1. Text may be entered directly from the keyboard. This occurs if 'c' is specified as a number and 'fileid' is omitted. 'c' lines are read from the keyboard and printed left-justified, as-is. Then, the rest of the letter is printed. This method is useful when you just want to insert a simple salutation in an occasional letter.

Example:      **;RD 3**

2. Text may be read from a file in the same manner as from the keyboard. This occurs if 'c' (as a number) and 'fileid' are both specified. A fixed number of lines must be provided for each entry in this case, even if some lines are blank. This approach is useful for personal lists to friends, where just about all sets take three lines.

Example:      **;RD 3,xmas/lst**

3. If 'c' (as a symbol such as '@') and 'fileid' are both supplied, the number of lines in each Name and Address (N&A) set can vary, so a mix of three, four, and five line entries is acceptable. When you use this form, each N&A set in the file must begin with a "Selection Code" line. The first (or only) character on this "code" line must be the character 'c' (we don't mean you must use the letter 'c'; it can be a '#', '&', '@', etc, ... anything you choose is O.K.). Each N&A group therefore ends at the next 'code line', or at the end of the file (for the last group only). Since the format allows a variable number of lines, blank lines aren't needed for short entries.

Example:      **;RD @,prospect/lst**

4. The fourth kind of list is like the third, but allows up to 99 variable names and/or phrases that can be inserted within the body of the text based on symbols that you place in the original form letter. This is the most flexible Form Letter support in ALLWRITE, since the merged text can be formatted as though it was an integral part of the letter, including underlining, right-justification and any other formats supported for your printer. Since a person's name can appear in the middle of the letter, on a fully formatted line, the result looks as though it was prepared just for the person receiving the letter.

Example:      **;RD @,customer/lst**

We will show you detailed examples of methods 1, 2, and 4 at the end of this section.

### Suggestions

The Form Letters feature of ALLWRITE can be very useful in many situations, and it's tempting for a new user to jump right in and attempt to try it before mastering the fundamentals of ALLWRITE. We suggest you become comfortable with the fundamental material in Chapter 3 before doing much with Form Letters.

Because Form Letters can be a confusing thing to grasp, it may be worthwhile right now to skip ahead and look at the examples. They show prototype letters and matching lists. After you've looked them over, come back to this discussion and continue reading.

### How Form Letters Work

When ";RD" is encountered, a control break occurs and all text preceding the control word is formatted and printed. If the first kind of list (from the keyboard) is used, the Formatter will prompt you to:

> > > **ENTER n LINES < < <**

and for each specific line will prompt:

**n: ?**

'n' will be a specific number, beginning with "1". You should type exactly what you want printed, and then press "ENTER". If several lines are to be read, you will be prompted repeatedly, and should enter one text line each time. If you had set the number of lines to "5", but only need three lines in a particular letter, just press "ENTER" in reply to the fourth and fifth prompts: that will leave two blank lines in the letter. If you specified '0' (zero) as the number of lines, then you will be able to type in a variable number of lines, ending with a "null" line (just ENTER with no text). You can include blank lines by typing a space and then ENTER: that's different from a "null" line.

The lines you enter are NOT joined to each other, nor to anything else: they are printed as-is.

If a number of lines and a 'fileid' are both given, the Formatter will make sure the file actually exists. Then, it will read the first 'n' lines from that file and print them as-is, and print the rest of the letter. This is the second kind of Form Letter. When the end of the letter is found, a page eject will occur, the letter will start over again, automatically and with no operator intervention. The next time the ";RD n fileid" is encountered, the next 'n' lines from the same file will be read and printed, etc. ALLWRITE will keep track of its position in the Name and Address file, and repeat the Form Letter until it reaches the end of the N&A file.

If 'c' (which may be any character that isn't a number and won't appear as the first letter of a name or address) is specified, a variable number of lines is read from the file. The rules are used for the third and fourth kinds of mailing lists are as follows:

1. Each entry in the N&A file must begin with a 'code' line, and the first character of this 'code' line must be the character specified by 'c', which should be a special symbol such as "@", "#", "&", etc. If you're also using tabs, make sure you don't pick the tab character for this!
2. The 'code' line may optionally contain a series of selection codes of your choice. These can be one-letter codes, or words separated by blanks. The codes can be used to selectively print only certain names and addresses.
3. When the N&A file is accessed for the first time, you will be asked to enter the Selection Code for this particular print run. You may enter any character string you wish, or just press "ENTER" to indicate that all entries in the file should be included into the letters that are about to be printed. This question is asked only once, and the code you enter will be used for the rest of the run.

#### **ENTER SELECTION CODE:**

4. If the 'code' line contains the Selection Code, or if no Selection Code was given, then the next several lines of material from the N&A file will be read and printed as-is, left-justified, but subject to the current settings of ";IN" and ";LM".
5. If the 'code' line does not contain the Selection Code, the set that follows it will be skipped, and the N&A file will be scanned until another 'code' line is encountered. Then, Step 4, above, will be repeated. This will continue until the end of the file is reached. N&A sets whose Code Lines don't match the criterion for this run will not be printed.

6. Once a name and address set has been selected for printing, all lines containing only text, up to the next 'code' line (or end of file), will be printed. When the next 'code' line (or end of file) is found, control reverts to the next line of input text from the letter itself. A separate letter will be printed for each qualifying entry in the N&A file. When the end of the N&A file is reached, the final letter will be completed and printing will stop.
7. You may include variable information in any or all N&A groups. This information is stored until needed, then printed within the body of the letter. Up to 99 variables may be defined within each group, and if none are defined for a given N&A, ALLWRITE will print a letter anyway, but ignore the signal for printing the variables. This variable information is not printed along with the normal N&A lines, but saved until referenced within the letter. You can create a N&A file that contains only variable information and no fixed N&A's: only the variables will be printed, and they will print only where you specify. This is the method that gives the most "customized" look of all to a letter. Any variable information that isn't referenced in the letter will just be ignored, so you can use the same N&A list for different letters. You can also carry the same information as more than one variable by repeating it.

#### **Defining Variable Information in a List**

To define variable information, use the code character twice, followed by a number from 1 to 99, followed by one blank, followed by the information you wish to substitute. Exact positioning must be used (no blanks after the double code character, exactly one blank after the number). For example, if '@' is the code character (which it is by default):

@ @1 Mr. Smith

would define "Mr. Smith" as variable #1 for this copy of the letter being printed. The double "@ @" will not conflict with the normal use of emphasis marks.

8. There is one 'permanent' or 'global' variable: "@ @0" ("@ @" would be whatever code character you've used, but the zero is what we are discussing now). The substitution value for "@ @0" always comes from the keyboard, not from the N&A file. You will be prompted to enter it on the first letter only and its value will be re-used for all subsequent letters printed during the current run. The main purpose of this variable is to provide a date in the letter, although any other similar use would be acceptable.

The prompt message reads:

**ENTER SUBSTITUTION VALUE (DATE):**

The ";CC" control word does not produce customized Form Letters and should not be used with ";RD". If you use "CC m" and ";RD n fileid", you will get one complete set of letters, and ";CC" will be ignored.

";RD", not ";IM", should be used to create Form Letters from a mailing list. ";IM" is intended for imbedding text from another file, and would be a very cumbersome way to do mailing lists. A Form Letter can imbed a logo or stock paragraph, of course.

#### **FORMATS OF A NAME AND ADDRESS FILE**

Until now, we've been describing what goes into the "letter" portion of Form Letters. Now we will describe what must be in the list of names and addresses. ALLWRITE can use either of two formats of mailing lists: a fixed number of lines for each N&A; or a variable number of lines, each preceded by a 'code' line. Each of these two formats is described below, and both can be produced by "MARVELIST", a separately available list manager intended for use with large lists.

**Fixed Number of Lines**

ALLWRITE expects a very simply structured Name and Address file. It must be a sequential ASCII file (the kind of file produced by your own BASIC programs or the Editor), with each entry created by BASIC'S "PRINT #n" statement. This kind of data is read correctly by the "LINE INPUT #n" statement. Example #2, later on, illustrates this format.

When the fixed number of lines method is used, each Name and Address within the file must occupy the same number of lines as all the others, even if some of these lines are left blank. This number must be the same as "the value of" 'n' specified in the ";RD n fileid" control word used in the Form Letter itself.

If end-of-file occurs unexpectedly in the N&A file, extra blank lines will be printed for that final Form Letter; no special error message is given.

If "fileid" is not found on an on-line diskette, this message is displayed, and you can type in another name or insert the correct disk to try again:

**FILE NOT FOUND. ENTER FILE NAME (DEFAULT=name):**

Name and Address files may be created easily using the editor or many other programs. If you are using a Mailing List program that carries its data in a different, incompatible format, it will be necessary to write a simple basic program that creates a new N&A file that can be read by ALLWRITE. Certain "Data Base" managers can create lists that are exactly or nearly in the form needed by ALLWRITE: Maxi-Manager does this, and so does our own MARVELIST.

**Example #1: - Form Letter with Keyboard Input**

The letter might look something like this:

```
;in 30
;fo off
Box 560
No. Hollywood, CA 91603
;cm get the date
;rd 1
;in 0
;sk2
;cm get the Name and Address
;rd 6
;cm print the body of the Form Letter
;sk;fo on
    Thank you for your donation to/ the library ... (etc.)
```

If you run this through the Formatter, you will be prompted to enter one line (the date), and then to enter six lines (the Name, Address, and salutation, if that's what you wanted printed). If you wanted to run several letters, you would have to re-run the same file again and again.

**Example #2: Form Letter with Fixed # Lines from Disk**

For the second example of Form Letters with a fixed number of lines, we will use a Name and Address file. Each N&A occupies four lines. The letter might be very similar to the above, except for the omission of the date. You could still read the date from the keyboard each time, but ALLWRITE will not remember what you enter from one letter to the next, since you might be entering variable information such as amounts due.

;rd 4 GOODGUYS/NA  
 We appreciate your recent order, and hope that ...

The "GOODGUYS/NA" file should look something like this:

J .R. Jones  
 1234 Oak Street  
 Palm Springs, CA  
  
 Ms. Jennifer Leeds  
 Acme Widgets, Inc.  
 1555 Main Street  
 Los Angeles, CA  
 Harry Hopkins  
 4777 Hollywood Blvd  
 Suite 4  
 Hollywood, CA

<==== blank line

Notice that the very first entry (to J. R. Jones) only needed three lines, so a blank line was entered to fulfill the "4" line requirement. If some entries in the N&A file had to be more than 4 lines, then all entries would have to be changed to meet the maximum requirement.

Example #3 - Variable Number of Lines with Substitution

This final example uses the fanciest features of Form Letters: variable number of lines, text substitution, and selection based on codes. This is the fourth type of Form Letter, and could be simplified to the third type by changing "Dear @ @ 1" to "Dear Member", and removing the "@@1" within the body of the letter.

Here is a sample letter and three entries from the list:

-- PROTOTYPE Form Letter --	-- N&A LIST --
;LL 40;im logo	@A,VP,7/84
;RD @ club/lst	Renee Desmond
;sk	123 Oxnard St.
Dear @ @ 1,	No. Hollywood, CA
;pp	@ @ Renee
The Valley Chapter of	@X,6/85
"Computer Addicts" will	Ron Rogers
hold its next meeting on	MagiPrint, Inc
Saturday, April 14th, at	Box 5544
the Victory Inn. @ @ 1,	Van Nuys, CA 91607
the meeting should be	@ @ 1 Mr. Rogers
particularly interesting	@A,12/83
because we will have	Mike Salven
@\$no one@% talking about	Open Roads Bikes
Word Processing. See you	8811 Vanowen St.
Saturday!	Suite 3B
---- LOGO FILE ----	Sun Valley, 92005
;CE ON	@ @ Mike
COMPUTER ADDICTS	
123 Chip Place	
Silicon Valley, CA	
;ce off;sk2	

When the letter is processed by ALLWRITE and the ";RD" is encountered for the first copy, the at-sign "@" will be picked up as the 'code' line marker, and you will be prompted:

**ENTER SELECTION CODE:**

If you entered "A" (upper/lower case doesn't matter: A=a), then letters would be printed for the first and third entries in the N&A file (Renee Desmond and Mike Salven). The second entry doesn't have the letter "A" in its code line, so it won't be printed.

The variable defined for each of these entries ("Renee" and "Mike") would replace the "@ @ 1" in the salutation of each resulting letter. However, if "@ @ 1" were omitted from the letter, the variable would be ignored. If "@ @ 1" occurred more than once in the letter, the substitution would occur as often as needed. If several variables were defined and also occurred in the letter, substitution would occur as needed.

If one of the codes happened to be an expiration date (5/84, for instance), it would be possible to send out "dues notices" from a list like the ones shown, since the 'code' lines contain such dates.

The first letter generated by these files would look like this except for the line length, which has been reduced to fit this page:

COMPUTER ADDICTS  
123 Chip Place  
Silicon Valley, CA

Renee Desmond  
123 Oxnard  
No. Hollywood, CA. 91604

Dear Renee,

The Valley Chapter of "Computer Addicts" will hold its next meeting on Saturday, April 14th, at the Victory Inn. Renee, the meeting should be particularly interesting because we will have no one talking about Word Processing. See you Saturday!

**OTHER FORM LETTER FEATURES -- RD Control Word**

The capabilities of ";RD" we've just discussed will be more than sufficient for most people. However, there are two more things you can do with this control word. The full definition of ";RD" is shown below; as you can see, it has two extra, optional capabilities:

**;RD [ c ] [ ,m [ ,n ] ] [ ,fileid ]**

'c' is the code symbol used to separate different names in the mailing list, and also used for variable substitution within the letter. 'm', which we didn't mention before, can be used with 'c' to define the number of lines that will be printed when ";RD" is used: if there are fewer text lines in a particular N&A set, blank lines will be added after them so that 'm' lines get printed; if there are too many lines in a set, the last ones will be discarded. This is useful when filling out pre-printed forms that have boxes or specified positions for certain entries.

If 'm' is used without 'c', then the N&A file must use fixed length entries of length 'm'. The ";RD" will read 'n' such entries at a time.

'n' is used to print side-by-side mailing labels, and its use will be explained soon. "fileid" is the name of the list.

**HIGH VOLUME FORM LETTERS -- ";FL"**

If you need to print a short (two pages or so) form letter, you can reduce the number of times the disk must be used by including this control word near the beginning of the letter:

**;FL**

This stands for "Form Letter", of course, and tells the Text Formatter to read the entire letter into memory, just once, and then print all the letters using that in-memory copy. If the letter fits into memory, it won't be read from disk over and over as each name and address is inserted from the mailing list. This can save a little time, but more importantly, it reduces disk access, and also lets you remove the disk containing the letter so you can insert the "list" disk.

If the letter is too large to fit into memory, if it uses fancy features (multiple columns, for instance), or if it uses any ";IM" (imbeds) or ";AP" (appends), then the disk resident copy of the letter will be used instead automatically.

The ";RD" control word must also be used in its normal place in the form letter: ";FL" does not replace ";RD", but only tells the Text Formatter to print from memory if there is sufficient room to do so.

";FL" must appear near the beginning of the letter if it is to be effective. It must follow any ";XT" control words, but should precede anything else. (Comment lines don't matter.)

This control word can be used as a run-time option with any small file, even if it isn't a Form Letter. If the file fits into memory, you can reduce the time your disk drives run.

**MAILING LABELS**

Personalized Form Letters are very nice if you have the time to print them, but if there are hundreds or thousands of names in your list, it will be impractical to print a custom letter for each of them. When your list grows large enough, you will be far better off printing mailing labels instead.

ALLWRITE can print labels from exactly the same lists as those used for Form Letters, but there's no "variable substitution" capability, so the "@ @ 1" approach cannot be used. However, both the fixed number of lines and the variable number of lines methods will work, and since the Name and Address portion of a list is printed on a label, that's all that's needed.

To make Mailing Labels as easy as possible to use, we've included a standard setup in "ALF/DEF". It will produce 3-up (side-by-side) labels, each of which is one inch high by 2.6 inches wide. Thirty labels are printed per "page" with this setup, and these are sizes that correspond to AVERY 5375 /5380 labels. "ALF/DEF" can be modified with the editor, so you can create a different setup to match other kinds of labels. To use this standard setup, you will need a list, which you can create with the editor or a List Manager.

When you run ALLWRITE and are asked for the document name, just use the name of the setup file itself:

**ALF/DEF**

The "labels" setup is the first thing in it. When you are asked to enter a "substitution value", reply with the name of your mailing list. After you've answered that question, the labels will be printed. That's all there is to it.

If the standard setup meets your needs, you don't need to know how it works, but if you want to change it for some other kind of label, the rest of this section will show you how easy it is to do so.

;pp

At the end of our discussion of Form Letters, we mentioned that there's an extra, optional number, 'n', that can be used with the ";RD" control word. 'n' tells ALLWRITE to repeat the ";RD" a certain number of times, so that it will process several sets of names and addresses together. If the labels you are using are arranged three across and ten down, there

will be thirty labels to a page, so 'n' should be '30' in this case. The number of lines needed on each label must be fixed, and the optional number, 'm' is used to define that value. ('m' is the number of lines per label; 'n' is the number of labels per page.)

Now, although we haven't put all this together for you yet, the only thing missing is a way to print these 30 labels in ten rows of three across, rather than one after another down the page. That's done by printing the labels in multiple columns, as shown in the example below.

### **Default Labels Setup**

To understand how these controls are used, let's look at the actual default setup in "ALF/DEF":

```
:LB MAILABEL
:FL          ;cm keep this little layout in memory
:CS          ;cm using cut sheet labels
:TM 0;BM 0;LM 0 ;CM eliminate all margins
:LL 79;CB 3;23 ;cm 3-up labels 2.3" wide each, 0.5" gap
:RD @ 6,30 @ @ 0 ;cm 6 lines/label, 30 per page (10 down)
:EN MAILABEL
```

";CS" lets us use cut sheets, since these particular labels come that way. The Formatter will stop at the end of each page so you can insert another sheet. We don't want any margins, of course, so the three "margin" control words are all set to zero.

";CB 3,23" causes 3 column printing, with each column 2.3 inches in width. That uses 6.9 inches, and the line length was set to 7.9 inches, leaving us with one extra inch. ALLWRITE divides this unused width evenly between the columns, so there will be one-half inch of space between each label (that's called a "gutter"). That's all there is to multiple columns, by the way: it's really simple to understand and use.

The ";RD" control word selects the at-sign as the code symbol for the list; specifies six lines (exactly) for each label, which matches the one inch height of these labels; and thirty of them to be printed on each page. Since they will be printed in three columns, there will be 10 rows of labels on each page.

Instead of a normal "fileid", this ";RD" uses the special substitution variable, "@ @ 0", so the first time the Text Formatter finds that, it will prompt you to enter its value. You should reply with the file name of your mailing list.

If you have two-up labels, each 3.5 inches wide and one inch high on continuous form page, the setup should be:

```
;FL;TM 0;BM 0;LM 0;PL 6
;LL 70;CB 2,33
;RD @ 6,2@ @ 0
```

That will leave a 0.4" gutter. It is the setup we use for our own mailing labels, which are on continuous form, pin-feed, fan-folded paper. The short page length (6 lines) has room for only one pair of side-by-side labels per "page", but since the margins are set to zero and cut sheets are not in use, each row of labels constitutes a "page", and will be printed on the very next row of labels. This method uses as few labels as possible by not printing the last few names down the first column of labels, which would waste the second column.

### **Continuous Forms and Cut Sheets**

Our default setup uses cut sheets containing 30 labels each. Actually, there are 33 labels on each sheet, but if you're running in cut sheet mode, your printer probably cannot push the last inch of paper up to the print head, so the last row usually can't be used. If you're running with continuous forms, you will want to remove the ";CS" from the setup.

**Note to NEWSCRIPT Users**

ALLWRITE doesn't support NEWSCRIPT's "Super-Lists" because the Editor can store about 400 name and address sets in a single file. If you have bigger lists, you can use the DOS "APPEND" command to create a separate, large file; or run each list separately; or use a full-scale List-Manager instead of the Editor.

**LEGAL DOCUMENTS -- NU Control Word**

This topic covers the different ways in which printed lines can be numbered. The main use of this is in the preparation of contracts and court transcripts, but it can also be used for printing screenplays and poetry.

A pre-defined setup for legal transcripts is included in "ALF/DEF". If it meets your jurisdiction's requirements, all you have to do to use it is start each transcript as follows:

**;IM ALF/DEF,LEGAL**

That's all you have to do, and the rest will be done for you. These are the default settings that will be in effect:

;NU 2,3	;cm 2=number every other line
;cm	3=left margin following numbering column.
;cm	To number all lines, set 1st value to 1.
;TM 7,2	;cm leave 7 lines in top margin above body
;BM 10	;cm leave at least 10 lines at bottom
;LM 9	;cm left margin preceding numbers
;LL 60	;cm set line length to 60 characters
;LS 2	;cm double space the body
;PI 10	;cm use PICA pitch (10 chars per inch)
;JU OFF	;cm don't right-justify the text
;TT//\$/	;cm place page number in upper-right corner
;PP7,0D	;cm 7/10" paragraph indent, but not now

You can change any of these to meet your local requirements, or add additional control words if necessary: the capabilities of the ";NU" control word will be explained shortly. Notice that there are two left margins in effect when you use ";NU": the one to the left of the numbers, and the one between the numbers and the body of the transcript. ";LM" sets the normal left margin to the left of the numbers, while ";NU" sets the margin between the numbers and the body.

Note to NEWSCRIPT users: The margin controls here are the reverse of what they were in NEWSCRIPT. If you've been imbedding "LEGAL/SET", all you have to do is modify it to match this change, and then you can keep using it: if you don't have to start using "ALF/DEF,LEGAL" or change any of your existing transcripts.

**Question and Answer Indications**

What we've just covered is sufficient for contracts, but not for transcripts. In a transcript, the questions and answers are highlighted, often by a deep indentation. You can define a pair of Editor "Soft Keys" (see Chapter 4 for the mechanics of doing this) to stand for each of these repeated prefaces. Soft key "X" can be used for the question, and "Y" for the answer. Here is a pair of suggested definitions, with control keys shown according to the convention we've been using throughout this book:

<N>;pp<N>Q: <Q>  
<N>;pp<N>A: <Q>

<N> means "Control N", and opens a blank line. ";pp" starts a new paragraph, using the indentations specified in the defaults file. <N> ends that control word and opens another blank line. There are 6 spaces after the "Q:" and after the "A:", so there will be that many blanks after those letters. (The paragraph indent will occur to the left of them.) If right margin justification is off ";JU OFF" then this is all you will need. If you want the right margin to be justified, you should use 6 hard spaces instead of 6 regular ones: hard spaces don't get padded.

Once the soft keys have been defined this way and tested, they should be saved to disk, probably in the main defaults file:

**CMD=> KEY SAVE AL/DEF**

Thereafter, each time one of them is pressed, the definition shown above will appear on the screen and the cursor will be positioned just after the 6 spaces, so you can just continue typing the text. Pretty easy, Yes?

**OTHER FORMS OF LINE NUMBERING**

If you need to change the values assigned to ";NU", or want to use its other capabilities, the full definition of this control word is:

**;NU [ n1 [ ,n2 [ ,n3 [ ,n4 [ ,C ] ] ] ]**  
(or)  
**;NU OFF**

Before we start explaining this one, please recall that the square brackets indicate that values are optional; the brackets themselves are never typed in. However, the commas are needed.

Every 'n1'th line will be numbered, and the default is '2', meaning "number every other line of the page." This doesn't refer to printed lines, but to physical lines on the page. If you are double spacing (;LS 2), each printed line will be numbered. If you are single spacing, every other printed line will be numbered.

'n2' is the gap to be left between the numbers and the body of the text. It is expressed in tenths of an inch, and the default is '8' (8/10").

Each printed number will be 'n3' bigger than the one before it. The default is '1', but if you want the lines to be numbered 2,4,6 ... then this value should be '2'.

The Text Formatter assumes the numbers will not exceed '99', so it allows two print positions for them. If you will be numbering above '99', specify the number of digits as 'n4'. For example, if you want to number up to 250, 'n4' would be '3' (three digits in "250").

Normally, the line count is reset at the beginning of each new page. If you want the numbering to continue increasing across page boundaries, include the letter "C" (for "continuous numbering") after the last of the numbers.

It might be appropriate to give an example. If you want to number every fifth line, want each line to have its actual number (5, 10, 15, etc.), want the numbering to continue increasing from one page to the next, and want the numbering to be one inch to the left of the text, you would specify:

**;NU 5,10,5,3,C**

The first '5' specifies that every fifth line should be numbered; '10' leaves a 10/10 of an inch middle margin; and the second '5' is the increment added to the counter each time it gets printed. The '3' reserves three print positions for the numbers, so up to '999' lines (15 pages or so) can be numbered with the same size margin.

To turn off line numbering: **;NU OFF**

**CREATING A TABLE OF CONTENTS -- TC Control Word**

A table of contents consists of a series of lines, each of which contains a topical entry and the page number on which it begins. There also can be some dots between the words and the page number, but that is a matter of style, not a requirement.

To create a table of contents, all you have to do is include this control word, followed by the topic name, throughout your text:

**;TC your words go here**

This must be the last control word on the line because everything through the "ENTER" symbol will be included in the table of contents entry. ALLWRITE will store 'your words' along with the current page number, and then print the accumulated entries afterwards. The default setup for formatting the table of contents is in "ALF/DEF", starting at the label, "TC". If you want a different format, you can just change the definition in the defaults file, since ALLWRITE always uses that particular definition when printing the table of contents.

The entries are stored on disk, and when the document is completely processed, you will be asked whether or not you want the table of contents to be printed. This gives you a chance to not print it. Please note that a large table of contents requires a large amount of disk space, and the disk cannot be removed while the main document is being processed. So, if you have only two disk drives and a document that spans several diskettes, the table of contents must be written to drive 0, since the diskette in drive 1 will change from time to time.

What you type after the ";tc" will be printed exactly as you entered it: One space is removed after the ";tc", and any additional spaces are kept; that's how we produced the indentation in our table.

If you don't type anything after the ";tc", a blank line will be placed on the printed page. That line will not have a page number on it, as you can see in the Table of Contents in this book.

If you want a row of dots to be used as "filler" between the words and the page number, you can use the ";TB x nP" control word, where "P" tells ALLWRITE to pad with "periods". That was explained in chapter 4 under "tabbing" and is included as the default setup within "alf/def". If you want the space left blank, remove the "P" from the tabbing specification.

The file "ALF/DEF" contains a default layout for printing the Table of Contents on 8-1/2 by 11 inch paper. This file must be on-line when the T/C is printed. You may want to modify the file to meet your needs, or make a second version of it for 5-1/2 by 8-1/2 inch formats.

**SUPPRESSING THE TABLE OF CONTENTS**

When printing rough drafts, you may not want to take the time to print the table of contents. The ";NO" control word can be included in your files or as a run-time option to suppress the collection and printing of that material, as explained earlier in this chapter:

**;NO TC**

**DELAYED PRINTING OF TABLE OF CONTENTS**

If you don't want to print the table of contents immediately, but will want to print it later on, you can reply "NO" when asked whether it should be printed. The collected material will remain on disk, and the files will have the same name as your original document file, but will have the extension "/TCT". For example, if the first file of your document was called "INTRO/CH1" (that's how this book started), the table of contents file would be:

**INTRO/TCT**

It will remain on disk until replaced by another print run, or until you remove it. If you want to print it later on, you can just tell the Text Formatter it's a normal file. It contains all the information that's needed to print it. Since the file looks like any other document, you can even edit it first, if you want to make some minor changes.

**Examples:**

```
;tc CHAPTER 1 -- Introduction
;tc   How to use this book
;tc   Equipment requirements
```

Those, of course, are the first few lines of the table of contents in this book. Each line was typed into the document just before the relevant topic began.

**CREATING AN INDEX -- IX Control Word**

ALLWRITE can produce an index similar to the one at the back of this book: words and phrases are placed in alphabetical order (sorted); multiple occurrences of the same term are combined and their different page numbers listed in order; references requiring more than one line are printed as hanging indents; and two-columns are printed per page. In other words, an index created with ALLWRITE looks like the index in any reference book, not like something put together from 3X5 cards.

The procedure for creating an index is similar to the one just described for creating a Table of Contents: each time something should be referenced by the index, it should be identified at that point in the text. Doing this well takes some skill, so after describing the mechanics of the procedure, we'll discuss methods of doing it effectively.

The control word used to identify an index reference is ";IX", and it can be followed by any one or more words or phrases that should appear in the sorted index. We'll used the word "term" to refer either to a single word or to a phrase, since the indexing facility treats them identically. If several terms are used with a single ";IX", they must be separated by semi-colons:

**;IX term [;term ...]**

The brackets, of course, aren't part of what you type: we used them here to show that extra "terms" may or may not be used in any particular index entry.

";IX" does not create a "control break", so the text just before and just after an index entry is printed as though the index entry hadn't even been there, but the current page number is associated with all the terms in the list. For example:

```
... unless disk drives are
;ix disk drives; maintenance; problems; breakdowns
checked regularly ...
```

The text will print as:

**... unless disk drives are checked regularly ...**

and four index entries will be generated.

";IX" must be the last control word on a control word line because everything after it, until the "ENTER" symbol is reached, is considered as an index entry. It doesn't matter whether you use upper or lower case, since the part of ALLWRITE that combines and sequences the terms will treat "A" and "a" alike. However, it won't necessarily pick the one you want for printing, so if "case" matters, you should try to be consistent.

If you want to index something that includes a semi-colon, it must be the only entry following ";IX", and the entry itself must be preceded by a semi-colon:

**;ix ;centering: ;CE**

The index entry would be "centering: ;CE".

The index program combines terms into a single reference entry only when they match exactly (except for case), so "car" and "cars" will be treated as two separate entries. This means some proof reading will be needed afterwards, but since the pages are given with the words (after all it's an index), it's fairly easy to go back through the original text and clean it up. If the same term is referenced more than once on a page, that page number will appear only once for that term when the index is printed.

Index entries are collected from the text as the Formatter does its processing, and stored in a disk file. When the Formatter is done, it asks you whether you want the index to be printed. If you reply "YES", the Formatter will run a separate program, "ALINDEX/CMD". It will read the collected entries, sort and combine them, create a second file, and pass that file back to the Text Formatter for final printing. The process of sorting and condensing the raw terms into printable form is very fast: It's limited only by the data transfer speed of your disk, since the sorting and condensing take only a couple of seconds, even on a very large index. Similarly, printing the result is a function of the Text Formatter, so even when printing in multiple columns, it will be limited only by printer speed.

Since "ALINDEX/CMD" is used to process the index, it must be on an on-line disk when printing of the main text ends, or the Formatter won't find it. If that happens, you can either insert the correct disk (without removing the one containing the index entries), or cancel the step for the time being and run "ALINDEX" by itself later on. If you do that, you'll have to tell it the name of the unprocessed index file, which will be:

**mainname/IX1**

where "mainname" is the name of the first file comprising the document you just printed. By the way, the final, sorted index will be given the name:

**mainname/IDX**

and is the file that the Text Formatter will use to print the actual index. For example:

<b>Document I.D.:</b>	<b>psych/ch1</b>
<b>unsorted I.D.:</b>	<b>psych/ix1</b>
<b>sorted index:</b>	<b>psych/idx</b>

The file created by the Formatter must be kept on a diskette that is not removed while the main document is being printed. If you have a large document that covers several disks, you will want the index information to be collected on drive 0 if you have only two drives, and there must be enough room for it. This requirement is similar to the one stated for a Table of Contents but an Index file generally is larger than a T/C file. In both cases, you must be very careful not to remove a disk onto which a T/C or Index file is being written, because the operating system will not show what you've done, and will write new entries in the middle of other files on the second disk. If you don't have enough disk space to store a T/C or index, either buy another disk drive or don't use these features!

A standard layout for an index is in "ALF/DEF", and that file must be on-line when the index is printed. You may want to modify that layout or create a second form of it for smaller paper.

### Maximum Size of an Index

There is an upper limit to the number of terms the index facility can handle: it's based on available memory in the computer, not on the number of terms themselves. If you happen to use the same average lengths of words and phrases we did for this book, you should be able to reference between 2,500 and 3,000 separate entries in your index. Note, however, that as duplicates are combined, a smaller number of separate, printed terms will result at the end. The Text Formatter cannot determine whether it has extracted more terms than the Index program can process, so the Index program may issue a message telling you that there are too many entries. We have not provided a way to circumvent this problem because an index that is too large for ALLWRITE would fill more than twenty full size pages in two-column format (over 30 pages of the size used in this book). We don't expect this to be a restriction for very many people.

### Suppressing Index Generation

If you don't want index entries to be collected by the formatter during rough draft stages, you can include the ";NO IX" control word in the document, in your default file, or as a run-time option.

To save time with a very large document, you may prefer to print segments of it on several printers at the same time, or to print sections of it at different times. If you follow either of those methods, you will want to use the ";PA" control word to get the right starting page numbers on the independent segments, but you will still end up with several separate index files. There's no easy way to combine them, but you can make one extra "print" run of the entire document afterwards, and specify ";PG 9999". No printing will occur, of course, but a full Table of Contents and an Index file will be created.

### Techniques For Making Index Entries

Creating a good index requires planning and some effort. An obvious temptation is to let the computer generate index references for every word it finds, but the result would be useless: You wouldn't want "the", "it", "for", etc., in an index.

The next idea along these lines is to give a computer program a word list, have it search through the document files, and insert references each time a matching word is found. That doesn't work very well either, because a good index doesn't just contain exact matches, but also synonyms, descriptive phrases, and generic topics. The example we used earlier for disk drive maintenance illustrates that situation. If "disk drives" had been in the word list, it would have been indexed, but the other terms, which a reader would be likely to use, would not have been indexed. Trying to use a list of words to be ignored, while all other words are indexed, doesn't work either: you still will have the synonyms problem, and you would be amazed at how many words don't belong in an index. Prepositions, simple verbs, and articles don't even begin to cover the situation: except for "synonym" and "index", none of the words in the last few sentences belong in the index for this book.

What follows is based on my own experiences over the past ten years in developing computer-assisted indexes for several books. I've done them with word lists and mainframe programs too fancy for a micro-computer, and I've done them by typing the index references into the text. And I'm doing the one for this book by typing them in, one at a time. It's more work, but the purpose of the index is not to make my life easier: it's to make your life easier when you have to use what I've written. The only way to do that effectively is for me to read through the text, try to anticipate the terms you might think of when you need to find something, and then to put all those words and phrases in as index entries.

The best method I've found is to include index references while writing the original text, and then while proofreading on the screen for continuity, phraseology, and completeness, to add more index references as the need for them becomes apparent. The two hardest things are identifying the topics and references that should be indexed in the first place, and then choosing synonyms and alternate forms that a reader might look for in the index.

**CREATING FOOTNOTES -- FF, FT, and FN Control Words**

ALLWRITE can print footnotes in a variety of ways: at the bottom of each page, at the bottom of each column, at the end of a chapter, or at the end of a book. You can control the way footnote references appear in the body of the page and the way footnotes print at the bottom of the page, but ALLWRITE will use standard defaults if you don't specify anything special. Footnotes can cross from one page to another ("running footnotes"); they can be numbered sequentially or marked with asterisks (or anything else you choose); and they can use super-scripts if your printer has that capability, or parentheses otherwise.

Before a document can use footnotes, the footnote feature must be activated by this control word:

**;FF**

That stands for "Footnote Format", and if you don't include any parameters, built-in defaults @+1@= will be used. These defaults are listed below, with the parameters used to override them. When ";FF" is encountered, the current page layout settings are stored, and it is those settings that will be used to print all footnotes, regardless of any subsequent pitch, margin, emphasis, or line length changes. So, you should be careful to place the ";FF" after the basic layout of your document has been defined.

Each actual footnote must start and end with this control word:

**;FN (to start, and) ;FN END (to end)**

The footnote itself goes on screen lines of its own, not on the ";FN" lines. Since a footnote does not cause a "control break" in the text, the words of text preceding and following the footnote will be printed one after the other, with only a footnote reference (the generated number) added between them.

---

1. There do not seem to be any generally accepted standards for footnotes, so we used the guidelines in Porter G. Perrin's Writer's Guide and Index to English, (Scott, Foresman and Company, Chicago, 1950) pp. 388-398.

---

Here is an example of a simple use of footnotes, using all defaults. The rows of dots show where we omitted some text:

```
;ff
.
.
...
... According to Schneckly,
;fn
"Causes of Discoloration,..."
```

**;fn end**

**this can be avoided by ...**

If your printer can do super-scripts, the body line will print like this:

**According to Schneckly,<sup>1</sup> this can be avoided by**

If your printer cannot do super-scripts, the body will print like this:

**According to Schneckly, (1) this can be avoided by**

The footnote itself will be printed after the last body line of text on the same page. A short line and a blank line will precede the footnote; three spaces will be printed to the left of the footnote number and a period and a space will be printed between the number and the footnote itself:

---

**1. "Causes of discoloration,"...**

**SPECIFYING FOOTNOTE FORMATS -- FF Control Word**

Before getting into this, a quotation from Perrin <sup>2</sup> is in order: "Any description of the form of footnotes makes their use seem harder than it really is." The man knows whereof he speaks, because the simple method of using footnotes given above is all you'll probably need. However, if you have to define some formats of your own, three control words are available for the purpose:

---

**2. ibid, p.395.**

- ;FT** Footnote Title, printed just above footnotes
- ;FF** Footnote Format
- ;FN** Footnote start, end, or delayed printing

";FT" is used like any other title: there can be up to nine Footnote Title lines, and they follow the same rules as do Top or Bottom titles. However, a page symbol, "\$", will be printed as-is, not replaced by the page number. If you don't specify anything, the default is to use a two-line title: the first is a short underline and the second is blank. The first Footnote Title is printed on the first line after the body of the text. ";FT" must be used after ";FF".

";FF" is used to control the format of the footnote references in the body of the text, and the format of the footnotes themselves:

**;FF [D] [,C] [,n] [/tp [/ts [/fp [/fs [/ ]]]]]**

"The use of this control word and its parameters should be intuitively obvious to the most casual observer. @+"3@= For those of you who aren't casual observers, we will explain further ...

All the values are optional, but ";FF" itself must appear before any footnotes or footnote titles. You can use ";FF" more than once if you have to change the format or footnotes, but this is rarely necessary. As we said earlier, ";FF" stores the current page and formatting layouts for use throughout all subsequent footnotes, so be sure you place this control word after those specifications have been established.

**D - Disk Storage**

The optional letter 'D' tells ALLWRITE to store footnote references on **Disk** until you indicate where the accumulated footnotes should be printed. It's used with ";FN D" when you want footnotes printed at the end of a chapter or book, rather than at the bottom of each page, which is the default.

---

3. Unknown Athenian philosopher, circa 375 B.C.

**C - Column Changes**

The optional letter 'C' only applies when footnotes are used along with multiple columns, the number of columns on a page is to be changed, and 'D' (disk storing) is not in effect. It tells ALLWRITE to print the accumulated footnotes in mid-page, with the columns that reference them, and then to continue printing on the same page in the new number of columns. If 'C' is omitted and there are stored footnote references when the number of columns is changed, the footnotes will be printed and a new page will be started. The situation is not likely to occur in the first place, but we've given you a choice as to how it should be handled.

**B - Bottom of page positioning**

The optional letter 'B' tells ALLWRITE to print the footnotes as close to the bottom of the page as possible, rather than a line or 2 below the body of text. This is useful when you are using "Turabian" style footnotes. Note: the form of Turabian that requires footnote numbers to start over from "1" on each page is not supported by ALLWRITE.

**n - Running Footnote threshold**

'n' is an optional number that sets the minimum size (in print lines) a footnote must be before it is considered to be a "running footnote". A "running footnote" starts on one page but continues on the next page(s) because there wasn't enough room for all of it on the first page. 'n' defaults to '1', which means that all footnotes are allowed to "run".

If a footnote uses fewer than 'n' lines, but cannot fit on the current page (because there isn't enough room for all of it), then the entire footnote, and the line of body text that references it, will be moved to the next page. For example, if 'n' is set to '5', and a footnote is 3 or 4 lines long, but there are only two lines left on the page, the footnote and the body reference will be moved to the next page, leaving a slightly bigger bottom margin than usual. In the same conditions, if a footnote is five or more lines long, then the first 2 lines of it will print on the starting page, and the rest will "run". Generally, you won't need to concern yourself with running footnotes, but you can control them if necessary.

A minimum of four lines must always be kept together on the same page: the body line that references a footnote, the first line of the footnote itself, and the two lines of the footnote title. If there are only three print lines left on a page when the reference line is reached, those three lines will have to be left blank, and the reference line moved to the next page.

I feel as though I'm trying to describe how to tie a shoelace: it's a lot harder to say it than show it. So, let's explain the rest of the parameters for ";FF" and then give some examples.

The remaining four parameters are also optional, and if used, must be bounded on both sides by a delimiter (a symbol not used in the values themselves). The slash "/" is shown as a suggestion for this, since it's easy to see and to press.

**tp - Text Prefix**

'tp' stands for "text prefix", and is the character(s) to be printed just before the number or symbol that ALLWRITE generates for you within the body of the text. If your printer can do super-scripts, this defaults to the super-script emphasis mark, "@ +"; otherwise, it defaults to a hard space and a left parenthesis "(" . If you want a hard space, a pause, or something else there, you can re-define the prefix. It must be preceded and followed by a delimiter, and if you want spaces to be used, you must include them in your definition.

**ts - Text Suffix**

'ts' stands for "text suffix", and is the character(s) to be printed just after the footnote reference. If your printer can do super-scripts, this defaults to a sub-script or a "cancel super-script"; otherwise, it defaults to a right parenthesis and a hard space:")".

**fp - Footnote Prefix**

'fp' stands for "footnote prefix", and is the character(s) to be printed at the start of the first line of each footnote. It defaults to three hard spaces.

**fs - Footnote Suffix**

'fs' stands for "footnote suffix", and is the character(s) to be printed to the right of the footnote number/symbol in the first line of each footnote. It defaults to a period and a hard space.

**Examples:**

**;FF**

uses internal defaults: not 'M', not 'C', 'n'=1  
 tp = super-script or '(',  
 ts = sub-script, cancel super-script, ')',  
 fp = three spaces  
 fs = period space

**;FF 3**

at least the first three lines of a long footnote  
 must print on the same page as the reference.

**;FF D/@ +/@ - /@ +/@ - /**

accumulate footnotes on disk; do not print them at the bottom of each page.  
 Print the reference to each footnote, and also the number preceding each footnote, as a super-script. On an Epson or Microline, use '@=' , not '@-' (include spaces).

The third control word for footnotes is ";FN" itself. You can use it in any of five ways:

<b>;FN n</b>	begin footnote, set number to 'n'
<b>;FN c</b>	begin footnote, set symbol to 'c'
<b>;FN</b>	begin footnote, use next number or one more replication of the symbol
<b>;FN END</b>	end a footnote
<b>;FN D</b>	print the footnotes stored on disk

**;FN n**

Footnotes usually are numbered in ascending order (1, 2, 3 ...), and ALLWRITE begins at '1' by default. If you want footnotes in each chapter to be numbered starting at '1', instead of continuing the numbers from the previous chapter, you can use the first form shown above to reset the counter. The control word also indicates that text for a footnote begins on the next screen line.

:FN c

If you want to use an asterisk, a dagger, or some other special symbol for a footnote reference, you can specify it instead of a starting number. If you do this, subsequent footnotes will not be numbered, but will use the same symbol until you reset it. The control word also indicates that text for a footnote begins on the next screen line.

:FN

If no number or symbol is used, and numbering is in effect, then the next sequential number will be generated for you. The number will be printed in the body and also to the left of the footnote itself at the bottom of the page. If a symbol is in effect, it will be printed again and replicated up to four times, if necessary: two stars for the second footnote, three for the third, etc. If more than five of the symbol are needed, an error message is printed instead. Use of symbols is not recommended unless you only need one or two footnotes altogether: numbers are more flexible and conventional.

:FN END

This marks the end of a footnote. Everything from the ";FN" preceding it to the ";FN END" is considered to be part of the footnote itself. The text after "END" is "body text" and is joined to what precedes the footnote.

:FN D

This tells ALLWRITE to retrieve all accumulated footnotes from disk and print them where ";FN D" occurs. It's used with ";FF D" to print footnotes at the end of a chapter or book, rather than at the bottom of each page. Subsequent footnotes are accumulated as a new set to be printed later on.

**Special Considerations with Footnotes**

Emphasis marks can be used within footnotes. ";SK", ";BR" and other simple spacing control words also can be used. Some control words cannot be used within footnotes, but we won't list them because the restrictions rarely occur.

At most 250 footnote lines can be stored for printing at the bottom of any one page, but you'll probably run out of memory before this limit is reached. If you have too many long footnotes, you can use the "Disk" option instead: Its only limitation is the space available on your diskette.

Footnotes can be used within multiple columns, and will be printed at the bottom of each column in which they are referenced. "Column balancing" will be turned off when this happens, but the columns will be pretty well balanced anyway, except on the last page. If you change the number of columns while there are stored footnotes for the page, those footnotes will be printed before the column change takes effect.

If 'C' was specified in the ";FF" definition and ";CB" was subsequently issued to change the column definition, then the next set of columns will be printed below the footnotes (if there is room for them on the same page). If 'C' was not specified, the next set of columns will begin on the next page, even if there is room on the current page. We don't recommend the use of 'C', because footnotes in the middle of a page will look strange; but printing them at the bottom of the page, using a varying number of columns would look even worse.

You can have several footnotes in a row, but each one must begin with ";FN" and end with ";FN END". The first ";FN END" would be followed on the next line by another ";FN" to signal the start of the next footnote.

**CHARACTER SUBSTITUTION AT PRINT TIME -- TR, SY Control Words**

Some printers can print special symbols that aren't on the keyboard of the TRS-80. Some printers may have features that aren't directly supported by ALLWRITE (graphics, for example). This section will present two control words that are useful in these cases. This is definitely an advanced topic, so unless you need to use these features and feel comfortable with them, we recommend you skip this section.

Two control words are available: ";TR" and ";SY". The first stands for "translate" and the second for "symbol". We will discuss ";TR" first, but ";SY" may be a better choice in many cases.

Simple character substitution can be done by using the ";TR" control word. We introduced ";TR" earlier, when non-standard daisy wheels and thimbles were discussed, but the control word has another capability that applies to dot matrix printers as well.

The control word can be used in three different ways:

**:TR c**

'c' is any letter A-Z, and "case" doesn't matter ("D" is the same as "d"). This form of ";TR" tells ALLWRITE to Translate all following text using table 'c'. Table 'c' may be one of the tables previously identified by the ";XT c" control word, or it may be the at-sign "@", which selects the internal translate table for a Diablo or Spinwriter. See "Disk Resident Daisywheel Tables" for further details.

**:TR**

If no parameters are used, all translation is cancelled. This can be useful in switching back to the standard characters of a non-rearranged print wheel, or in canceling changes made by the third use of ";TR", which is shown below. Since the internal translate table and all tables read from disk remain available, you can re-issue the first form ";TR c" later on.

**:TR n1,n2,n3**

Adds 'n3' to all characters in the range 'n1' through 'n2'. Three numbers are expected here. 'n1' and 'n2' are the ASCII equivalents of characters. For example, '65' is the ASCII value for the letter "A". 'n3' may be any number from '-255' to '255'. Positive values need not be signed, and you can use negative numbers when necessary. Numbers outside this range, or sums that are too large or small, will be truncated (modulus 256), and you won't get the right results: the range of possible resulting values is 0 to 255.

This use of ";TR" changes text characters just before they are printed. For example: the paragraph symbol "¶" on this print wheel is an ASCII "124", but on a Radio Shack daisywheel, it is an ASCII '175'. One way to print the "¶" on an R/S daisywheel II would be to use the Editor's "special symbol" control key (in fact, that's probably the best way to print it). A second way would be to choose a key that prints something you don't need, such as the '>' (above the period) and then to let ALLWRITE translate it to the "¶" at print time. '>' has an ASCII value of '62', while "¶" on the R/S printer has an ASCII value of '175', which is '113' larger than '62'. You could use '>' throughout your text and have it print as a "¶" by including this once, near the top of the document file:

**;tr 62,62,113**

'113' will be added to all characters in the range '62' through '62' just before they print. In this example, there's only one character in that range, so it will be printed as '62+113', which is '175'.

The reason for allowing translation over a range of values is to make it easier to access low resolution graphics on certain printers such as the original Epson MX-80 (the one without GRAFTRAX). The use of ";TR" shown below would allow on-screen TRS-80 graphics to be printed "as-is" on that printer:

**;TR 128,191,32**

TRS-80 screen graphics are in the range of '128-191', while Epson graphics are in the range '160-223'. Adding '32' to the TRS-80 values produces the Epson symbols. However, bear in mind that the last character on a screen line cannot be the "ENTER" symbol, '142'.

You can use ";TR" in combination with ";IM" (imbed) to reconfigure your keyboard over and over again. That is, you can develop a list of character translations, store the resulting file on disk, and then imbed it whenever you want to switch to the special characters it defines. For example, if your printer can print Greek letters and defines "alpha" as an ASCII '200', you could re-define an upper-case "A" so that it would print as 'alpha' instead, leaving the lower-case "a" as a Roman letter:

**;TR 65,65,135**

If 'alpha', 'beta', 'gamma', and 'delta' happened to be in numerical sequence on your printer and you wanted to use the capital letters 'ABGD' for them, you could do this:

**;TR 65,66,135  
;TR 71,71,131  
;TR 68,68,135**

Tables of this sort only have to be built once, and then you can just imbed them whenever they are needed, turn them back off with a simple ";TR" to revert to normal characters, etc.

Please remember that ";TR" takes effect only when text is printed (or previewed), not while you are editing. By contrast, the "Special Symbol" key, <:> takes effect immediately right on the screen.

#### **SENDING SPECIAL SYMBOLS TO THE PRINTER -- SY Control Word**

Any sequence of ASCII characters can be set "as-is" to the printer by using the ";SY" (symbol) control word. This can be done in three ways, as shown below. Misuse of this control word can cause your printer to print incorrectly, eject paper, or stop printing entirely until you turn it off and back on. It also can cause ALLWRITE to lose control of the printer, or at least become unsynchronized with its actual state. We've included ";SY" for use by experienced people who have some technical background. Please note that some DOS's will not transmit all ASCII characters, so if you are not using our printer driver, ";SY" may not always work.

**:SY n1,n2,n3...**

'n1', 'n2', etc., are ASCII numbers in the range 32-255. The corresponding ASCII characters will be sent to the printer as-is, printed in mid-line along with normal text, and counted by the right-justification routine. This form of ";SY" is useful when you want a few special symbols to be printed. If you only want to send one or two special characters, it's probably easier to use <:> instead.

**:SY @n1,n2,n3...**

This is similar to the first form, but the "@" sign tells ALLWRITE not to count the characters in calculating right-justification. This gives you a way of sending "escape sequences" to the printer in mid-line. It's useful for accessing features not supported by ALLWRITE's Text Formatter. However, since the Formatter can't know the effect of your intervention, it cannot adjust line or page positioning if your escape sequence changes anything.

;SY/  
n1,n2,n3,.../

This is different from the other two forms because the slash character "/" tells ALLWRITE to start a new print line, provide a left margin, and then convert your list to ASCII characters and send them to the printer. The following rules apply:

1. the list of characters must begin on the next line (an "ENTER" must follow the slash);
2. there may be as many numbers as you need;
3. they should be separated by commas;
4. they must be in the range 0-255;
5. they can occupy more than one screen line;
6. the entire sequence is counted as one print line;
7. characters other than numbers are ignored;
8. the list must end with another slash "/"

When the list ends, ALLWRITE will send a carriage return to the printer and count one line as having been printed. You can have any sequences you like in the list, but they should result in only one line of printing; otherwise, ALLWRITE will lose track of where it is on the page.

The third form of ";SY" is useful in printing high-resolution graphics, but development of the lists of ASCII numbers can be somewhat tedious. If you want to draw your logo this way, a good approach would be to write a BASIC program that will generate the ";SY /" followed by the appropriate numbers.

Some Text Formatters have used ";IIX" as a control word for this sort of thing. We chose to not use that control word because ASCII numbering is easier to understand than hexadecimal.

### **VARIABLES AND LOGIC**

This is an advanced, specialized topic, and in presenting it, we will assume the reader has some programming experience. If the meaning of the word "variable" as it is used in computer programming is unclear to you, please skip this section.

We've already covered two of the features in this topic: the ninety-nine variables available for Form Letters, and the labels used to imbed portions of the files. However, there are some other control words available:

**;SE ;AR ;IF ;GO ;LB ;EN**

Some of these were explained as part of the "Long Document" topic. Please don't make any assumptions about how these work. ALLWRITE is a word processor, not a computer programming language.

### **ASSIGNING VALUES TO VARIABLES -- SE, AR Control Words**

ALLWRITE recognizes 100 variables, each beginning with a double at-sign "@@" followed by a number from zero to ninety-nine. "@ @ 0" is different from all the others because its value is set only once, from the keyboard, the first time it is encountered. Its mostly used to set the date, but the "Mailing Labels" setup in "ALF/DEF" uses it to let you identify the mailing list. The material that follows applies only to the other 99 variables.

All variables are "null" initially. You can assign numbers or words ("strings") to them in three ways: from a mailing list read by ";RD" ";SE" ("set"); or ";AR" (arithmetic). All three methods can be in use at the same time, so you should be careful to keep the uses of your variables consistent. ";RD" was covered in the "Form Letters" topic, and the other two are covered below. The maximum length of any variable is 255 characters, and if you get too enthusiastic, you can run out of memory.

**Set a Variable to a Value --SE Control Word**

The ";SE" control word is used for two purposes: to assign a value to a variable, and to change the symbol used to identify variables. By default, the symbol is the at-sign "@", but you can change it this way:

**;SE ?**

where "?" is any symbol you want to use. This affects Form Letters also. The symbol is doubled in text ("@@") to let you use a single "@" as a normal character.

When ";SE" is used to assign a new value to a variable, that variable's number must follow the control word, and the double at-sign must NOT be used. The reason for not including the symbol is that the Formatter substitutes the current value of a variable whenever that variable is found in text, and when setting a new value, you certainly don't want the old value to be involved. However, you can assign a variable to a variable, as shown in the third example below.

The variable number must be followed by a delimiter (usually an equal sign) and the new value. (A comma or a single space may be used instead of the equal sign). Here are three examples:

**;SE 8=January**

after this, whenever "@@8" occurs in text, it will be replaced by "January"

**;SE 13=27**

after this whenever "@@13" occurs in text, it will be replaced by "27". (We told you it wasn't like BASIC.)

**;SE 4=@@13**

after this, "@@4" will be replaced by whatever had been assigned to "@@13" when the above control word was processed. If "@@13" changes afterwards, it won't change "@@4".

**Concatenation of Variables**

In this context, "concatenation" refers to the joining of two variables into a single variable, not to text concatenation as used by the ";CO" control word. To concatenate two variables, just use the two variable symbols one after another.

**;SE 5=@@2@@3**

The first character after the variable number ('5' in this example) may be a space, comma, or equal sign. Everything after that character, including spaces, becomes part of the definition of the variable.

If '@@2' is "ABC" and '@@3' is "DEF", then '@@5' will become "ABCDEF". You can do this with several variables at a time, and can intermix constants (normal text) with variables. We have not provided "INSTR\$" or "MID\$" functions.

**ARITHMETIC --AR Control Word**

This control word can be used to perform one arithmetic operation at a time. When the arithmetic is done, the starting value of the variable should be "null" or numeric, not string. In its simplest form, ";AR" can be used to number "Figures" (or anything else, for that matter):

**;AR 3**

will add "1" to the current value of "@@3". Then, you can use it like this:

```
;sk15
;ar 3
;ce
Fig. @@3
```

The first time you do this, "Fig. 1" will be printed; the ninth time you do it, "Fig. 9" will be printed.

The general form of ";AR" is this:

**;AR n O v**

'n' is a variable number between 1 and 99. It's current value is used as the first operand of the operation, and the result is assigned back to the same variable, replacing its previous value. Only the number, and not the "@@", can be used.

'O' is one of five arithmetic operators:

<b>+</b>	<b>addition</b>	<b>n = n + v</b>
<b>-</b>	<b>subtraction</b>	<b>n = n - v</b>
<b>*</b>	<b>multiplication</b>	<b>n = n * v</b>
<b>/</b>	<b>division</b>	<b>n = n / v</b>
<b>M</b>	<b>mod (remainder)</b>	<b>n = n M v</b>

'v' is another variable or a numeric constant. Examples of using ";AR" will be shown soon.

Only integer arithmetic is supported. Negative values can be used, and the range of values is -32768 to +32767. Overflows are ignored; division by zero yields zero (not an error); quotients are truncated, not rounded; and "mod" yields the remainder of the division of 'n' by 'v'.

**LOGIC (CONDITION TESTING) -- IF Control Word**

Two values can be compared according to any of the six standard operators:

**;IF v1 O v2**

";IF" must be the only control word on the line, since everything between the operator and the next "ENTER" is included in the second variable.

'v1' is a variable symbol or a constant; 'O' is one of the comparison operators shown below; and 'v2' is another variable symbol or constant. If the result of the compare is "TRUE", processing continues with the next line of text or control words. If the result is "FALSE", all text and all control words are skipped until another ";IF" control word is encountered or the end of the file is reached. In the latter case, processing ends and any "append" will be ignored. If ";IF" is used with no operands, a "TRUE" condition will be raised and processing will continue: it is a good way to end an "IF group".

These are standard operators:

=      <>      < >      <=      >=

Those stand for "equal", "not equal", "less than", "greater than", "less than or equal", and "greater than or equal".

"IF" commands are not stacked or nested: the next ";IF" or the end of the file terminates a false ";IF".

### Ignoring Text and Labeled Text -- GO Control Word

We covered these fully in Chapter 4, under "Long Documents";

```
;AP file,label
;EN label
;GO label
;IM file,label
;LB label
```

";GO" only searches forward in the same file, and if the label isn't found, processing will end. If the label is above the ";GO", you can use ";AP" instead of ";GO", appending the current file to itself and specifying a label. You can also develop loops by using ";AR", ";IF", and ";AP", and if you aren't careful, you can create endless loops, which can use a lot of paper.

### Example

Without trying to teach you computer programming, we will show you an example that uses some of these control words:

```
;SE 1=5
;SE 2,7
;SE 3 ABC
;SE 4=@@2
;AR 4 + @@1
;IF 4 > 9
;@@4
;GO BIG
;IF
too small
;LB BIG
This always prints.
```

As shown, this would print "12 This always prints." If the first ";SE" had assigned the number "2" to variable one, the first ";IF" would have been false, and "too small This always prints." would have been printed.

This completes the discussion of variables and logic, and also ends the presentation of the Text Formatter.



## CHAPTER 6

### PRINTER SETUP

This chapter explains the special considerations involved in using certain printers with ALLWRITE or a TRS-80. Some printers, such as the ones sold by Radio Shack itself, just need to be cabled to the computer. Others have small switches that control their features. Often, these are inside the printer and hard to reach. The manual for your printer should explain how you access these switches, and we will not cover that here. If your printer doesn't operate properly with ALLWRITE, even though it is one of the supported printers, you probably just need to change some switch settings. The settings shown here can be used even when you are not using ALLWRITE: with very few exceptions, they will not affect how your printer operates under BASIC or anything else.

If your printer is supported by ALLWRITE, but not covered here, it means there are no switches to set or that the normal settings can be used. However, printer specifications are subject to change, so if your printer's switches don't match what we've described here, please refer to the printer's manual. If you can't decipher it, send us a copy and we will try to help you.

#### General Considerations

If your printer offers you a choice of whether or not to add a line feed to each carriage return, you will, in most cases, want that line feed to be added by the printer. This doesn't matter to ALLWRITE, since it can add the line feed for you; but it does matter to the rest of your programs, since they usually cannot add the line feed. A major exception to this rule occurs with certain Microline printers, as we will explain when describing those machines.

If your printer has a choice of "buffer" sizes, you usually will want to pick the largest buffer size available. The drawback is that the printer will continue printing after you've told ALLWRITE to quit.

#### SERIAL PRINTERS AND CABLES

ALLWRITE can send text to a serial printer much faster than the printer can print. However, most serial printers can accept text at "1200" baud (120 characters per second for us humans), even though they may be able to print only 30-45 characters per second. Also, most of these printers have the ability to tell the computer when to suspend data transmission and when to resume it. This is called "handshaking", and it is done in any of several ways: "ETX/ACK" and "DC1/DC3" are "software" handshakes, and both are supported by ALLWRITE. ("DC1/DC3" is also called "XON/XOFF".) Other handshaking methods use some of the hardware RS-232 control lines, and those are supported directly by the hardware of the TRS-80 or, in some cases, by your DOS.

If you are using a serial printer and have a choice of handshake methods, "DC1/DC3" is the best one to use; next best is "ETX/ACK". If you can use either of them you can run your printer at 1200 baud even though it cannot print that fast. If you do this, you will improve print speed significantly, especially when using proportional printing (it takes three control characters to print each text character).

If your printer has neither of these handshakes, but can signal "not ready" at the hardware level, you may still be able to run at 1200 baud. Otherwise, you must either run at a slower speed or try to use ALLWRITE's "padding" method. As a rule of thumb, if your printer can print 40-55 characters per second and you are using proportional printing at 1200 baud, you will need three or four pads. A problem will occur on long page ejects when padding is used: this does not occur with any of the other handshake methods.

**MAX-80**

If you are using a LOBO MAX-80, you must use the LDOS printer driver for parallel or serial printers. If you are using a LOBO MAX-80, you MUST use "printer attachment #1" for parallel printers, and "printer attachment #3" for serial (RS-232) printers. That is, ALLWRITE must control parallel printers directly, while LDOS must control serial printers.

**Serial Printer Cables**

A special cable is needed to use a serial printer with a TRS-80. If you're using a serial printer, you should already have such a cable, but sometimes these cables were not built correctly. If you cannot make this cable yourself, get someone to do it for you. We suggest you use a short "jumper" cable that can be used between your printer and the computer: It will eliminate the need to modify the main cable itself.

- switch pins 2 and 3
- switch pins 4 and 5
- switch pins 6 and 20
- all other required pins (1, 7, 8 etc.) connect to their own counterparts

When a pair of pins is "switched", it means that the wire from a pin at one end should be connected to the identified pin at the other end.

**SWITCH SETTINGS (Alphabetical by printer)****Ansdex****DP 9000, 9001, 9500, 9501:**

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Switch 1	O C C O C O O O
Switch 2	O O O O O O X X
Switch 3	C C O O O O O O O C

Note: "C" means "Closed" or "On"; "O" means "Open" or "Off"; "X" means "doesn't matter".

**BROTHER, COMREX**

There are at least two versions of the HR-1: One honors the escape codes in its documentation and the other one does not. If your printer does not do pitch changes correctly under ALLWRITE, try the "COMREX" setting instead; if that does not work either, your printer probably does not honor escape commands.

The Comrex printers all operate as described in their manuals.

The BROTHER HR-15, 25, etc., are Diablo-compatible and work properly when installed as a normal Diablo. This is also true of the Comrex. When using the "ANELIA" wheel, use our "ANELIA/TAB" with the ";wt" control word for proper spacing.

**C.ITOH****FP-1500 Starwriter**

1 2 3 4 5 6 7 8

Switch 1 (right)	O C O O C O C O
Switch 2 (left)	O O O C O O O O

**F-10**

1 2 3 4 5 6 7 8

Switch 1 (LEFT)	C O C C O C O O
	11 12 13 14 15 16 17 18 19 20
Switch 2 (right)	O C C C C O O C O O

**8510A**

1 2 3 4 5 6 7 8

Switch 1	O C O O O C O C
Switch 2	O O O O O O C O

Note: "C" means "Closed or "On"; "O" means "Open" or "Off".

**DIABLO 1600 Series**

To run a Diablo 1620 or other 1600-series printer at 1200 baud it is necessary to install a "jumper" on the "HPRO-2" board, near the back left hand side of the printer. This is described in the printer manual. Once you do this, the printer will only run at 1200 baud. However, since it can't print that fast, it will be subject to "overrun" (loss of text) unless you use special software. ALLWRITE is an example of that special software: it supports the "ETX/ACK" protocol of these printers and will not cause overrun at 1200 baud (or any other speed). Most of your other software will have a problem, however. The solution is to install the "jumper" in the form of a switch that extends outside the printer. If you are not used to making changes of this sort to your equipment, find a qualified technician to do it for you, or run at 300 baud.

Running a serial (RS-232) Diablo at 300 baud in proportional mode will be very slow. Running at 1200 baud will average about 30 characters a second, even though the printer is capable of 45 cps speeds. The limitation is a function of the transmission speed (120 characters per second) and the method the Diablo uses for proportional printing: it takes three control characters to print each text character. If you can run your printer at 4800 baud, it will print at full speed.

**Ribbon and Paper Changes**

If your printer doesn't stop when the ribbon (or paper) runs out, you can use ";PG" to print 100 pages or so at a time, and then check to see if a new ribbon is needed. With this approach, you won't have to check the printer every few minutes, and you won't risk printing blank pages for lack of ribbon or paper. When one page range runs out, ALLWRITE asks you for the next range, so you can print a large document in one pass without any problems. The method was useful in printing this book, since three ribbons were needed.

Epson

PRINTER	<-- SWITCH 1 -->	SWITCH 2
	1 2 3 4 5 6 7 8	1 2 3 4
Standard MX-80	C O O C C O O C	C C C O
MX-80 Graftrax-80	O O O O O O O C	O O C O
MX-80, Graftrax-Plus	O O O O O O O C	O O C O
Standard MX-100	O O O O O O C C	C C C C
MX-100, Graftrax-Plus	O O O O O O O C	O O C O
RX-80	O O O O O C C C	O C C O
FX-80, Buffered	O O O C O C C C	C O O C
FX-80, Proportional	O O O O O C C C	C O O C

Note: "C" means "Closed" or "On"; "O" means "Open" or "Off".

Some of these affect slashed zeros and buzzers, and you can set them however you like. If you're using an EPSON cable instead of a Radio Shack cable, and pin 14 is disconnected in that cable, then Switch 1-2 should be "Closed" (ON).

Right-justified proportional printing is possible on all these printers except the two marked "Standard". However, the method that must be used on the MX and RX series requires constant switching between text and graphics mode. This is very slow and, more importantly, may cause severe wear and tear on the printer. Therefore, we recommend against using Proportional except to confirm for yourself that you will never want to use it again. We must emphasize that this limitation is inherent in the printer, not in ALLWRITE.

If you want to try proportional printing on an MX or RX printer in spite of our warnings about the likelihood of damage to your printer, then be sure to specify ";SD 0,Y" so that all "padding" will be done between the words, not between the letters. Each time padding is needed, the printer must switch from text mode to graphics mode and back, so restricting the padding to be inter-word only will reduce the number of switches back and forth from 65 per line to only 12 per line. The print pitch will always be "16" (compressed).

ALLWRITE can print right-justified proportional text at full speed on the FX-80, with no special wear-and-tear, if you turn off its internal 2K buffer (switch 1-4 "OFF") and select "FX-80 Proportional" during installation. If you select "FX-80 Buffered" (switch 1-4 'ON') instead, then all the warnings for the MX-series will apply to the FX-series also. Regardless of how you install ALLWRITE for the FX-80, you must always use ";SD 0,Y" with proportional printing. No other "SD" setting will work properly. Also, as the FX manual states, backspacing will not work in proportional mode, and boldface cannot be used because the hardware prints proportional text in Emphasized Mode.

If you want to take advantage of the capabilities of GRAFTRAX to print a wide variety of fancy characters in different sizes, with full formatting control, we recommend the purchase and use of "DOTWRITER". It uses many of the same "control words" found in ALLWRITE, and is fully integrated as the character-graphics formatter for ALLWRITE.

Note that most Epsons do not have 12-pitch, only the GRAFTRAX-PLUS and later printers have sub-superscript capabilities, and only the FX-80 can perform reverse paper movements. Underlining on MX-series printers without GRAFTRAX -PLUS is done as a series of dashes under the characters, because those printers cannot draw continuous lines when in text mode. You can upgrade an earlier MX-80 to Graftrax-Plus by buying a new set of "EPROMS" from an Epson dealer, so you do not need to buy a whole new printer to get the fancy features.

Epson printers can perform "Emphasized" printing only when in 10-pitch. "Emphasized" corresponds to boldface level 1 in ALLWRITE. If you want to use boldface in 12 or 16 pitch, use boldface level 2 instead. Again, this is a printer restriction, and probably was intended to protect the printhead from overheating.

**MICROLINE**

Switches vary on this extensive line of printers. The following guidelines apply: use "8" bit code, not "7" bit; automatic line feed may be used with ALLWRITE, even on the 84, 92, and 93 (this is a change from NEWSRIPT's requirements).

There are two versions of the Microline 84: the original, called "Stage I", and the newer "Stage II". ALLWRITE supports both of these. If you have a Stage I machine, you can contact Okidata for information on upgrading to Stage II, but the upgrade is not needed just to use ALLWRITE.

**NEC****8023A**

1 2 3 4 5 6 7 8

Switch 1	O C O O O C O C
Switch 2	O O O O O O C O

Note: "C" means "Closed" or "On"; "O" means "Open" or "Off".

**Spinwriters**

There are several "lines" of Spinwriters (3500, 5500, 7700), and within each, there are two kinds of printers: the ones ending with zero, such as the 5530, use "native" Spinwriter control codes. The ones ending with a "5", such as the '7715', use "Diablo" control codes, but also use certain Spinwriter control codes when using "PS" (proportional) thimbles. The 3500 and 7700 printers have a switch that controls this, but the 5500 series does not.

ALLWRITE supports everything we've just mentioned. Depending on your printer, thimble, and thimble switch (if your printer has one), you may or may not need to use our internal translation table ";TR @" with a proportional thimble. Width control for these thimbles is built into ALLWRITE, so "PI 0" will work properly with the "BOLD PS" thimble, and "PI 1" will work properly with the "EMPEROR PS" thimble.

To use proportional printing, you may have to change some internal switches on some Spinwriter models. These changes will not affect printing when ALLWRITE is not in use, so you can set the switches once and forget about them. One critical switch controls the maximum number of characters the Spinwriter can receive from the computer before forcing an automatic "carriage return". If the switch is in its "factory" position, a carriage return will be forced after about three inches of printing, because control characters, as well as text, are counted by the printer!

While you are changing that switch, you may also want to set the buffer switches to the maximum size supported by your printer. On the 5500 series, the switches are in the back, on the right hand side, and very difficult to reach. On later machines, they are directly behind the control panel on the left hand side, and easy to reach.

The standard switch setting on some Spinwriters forces a carriage return after 133 characters have been received at the printer. This interferes with proportional printing (the printer counts control characters, even though it doesn't print them). and the solution is to turn "ON" switch "1-5" on the "G9NBA" board at the back of the printer. (The switch is hard to reach, and the cables are easy to break; if you don't feel comfortable about doing this yourself, find a qualified technician to do it for you.) Changing the position of this switch will not affect anything else you do with your printer.

### 5500 PROBLEM

NEC 5500's manufactured before 1981 have a factory defect that causes them to print the right-most character of a proportional line in the left margin, once or twice per page on a random basis. The serial number of the "good" board is "G9GLB-A2". Not all NEC representatives know about this problem, but the people in Field Engineering at Lexington, Massachusetts are familiar with the problem. There's only one way to correct the problem: replace the bad circuit board. This is not an ALLWRITE or computer software problem, and replacing the board has eliminated the problem in every case reported to us. The problem is less common in 5530s than in the serial models, but can happen even on a 5530.

### Customized Spinwriters

Many independent companies have modified or enhanced various Spinwriters and sold them under a variety of brand names. You must determine whether your machine functions as a "Diablo" model or as a "native" model. Also, if your printer can operate in both parallel and serial mode, you definitely will want to operate it in parallel mode: it's much faster and more reliable that way.

On RS-232 Spinwriters, a good hardware "handshake" to maximize print speed is available. A special cable, similar to the one described for "Serial Printers" earlier, must be used. It differs from the one already described in that pin 19 should be connected to computer pin 6, instead of connecting pin 6 to pin 20. If this paragraph was Greek to you, but you need the "Reverse Channel" handshake it describes, find a qualified technician who can build a cable for you. The wires in those cables are very thin and it's easy to break them if you aren't used to doing this sort of thing. If you have a parallel model of Spinwriter, this doesn't apply to you.

### RADIO SHACK

Radio Shack offers the most complete line of printers available. Most of them follow the same control code conventions, although proportional character widths vary from 1 printer to another. In general, all switches should be "OFF" (DOWN), unless you have a specific reason for using a different setting. The only switch you may need to know about is the one that controls "Auto Line Feed" and it should be set to allow a line feed whenever a carriage return occurs (that is the standard setting for TRS-80, but not for the rest of the world). ALLWRITE does not support the IBM capability mode of these printers, since it only runs on the TRS-80.

You can access the extra symbols of the DMP and DWP printers through use of ";TR" or the Editor's "Special Symbols" key. We support color on the CGP-220 through our color emphasis mark (see chapter 5); double-height on the DMP 430 through the double-height emphasis mark (but we do not support it on the DMP-110 because it is not practical to control the line spacing at the same time); virtually all pitches and character sizes on all Radio Shack printers; but we do not support "Extended Programming Mode" on the daisywheel printers, so we cannot support the 15 pitch wheels. We recommend use of Madeleine or Cubic proportional wheel for best results on the daisywheel printers; although the wheels are expensive, they are long lasting and give the best impressions of any daisywheels on the market.

ALLWRITE supports all fonts and pitches of the DWP 2100P, as well as downline loadable fonts (if you have that printer, you've already noticed how we took advantage of it in printing this book). Please see the ";DF" control word for use of loadable fonts, and contact us for pricing on the fonts we can supply.

## CHAPTER 7

### ERROR MESSAGES

Error messages are displayed when something goes wrong. These messages may be issued by DOS, The installation procedure, the Editor, the Text Formatter, or the index generator. The conditions causing some messages may be so minor that they really shouldn't be called errors in the first place. The serious errors are the ones associated with unreadable disks, because they mean that some of your text may have been lost.

Serious disk errors are rarely caused by errors in the computer or software: most of the time, they are caused by dirt, wear and tear of the diskette or disk drives that are adjusted incorrectly, or marginally defective diskettes whose flaws were not detected when the disks were formatted. A barely visible speck of dust can scratch a diskette (and possibly the disk drive head, which can cost over a hundred dollars to replace) and wipe out several characters of text. Unfortunately, text is stored by DOS in "sectors" of 256 characters each, so when it gets lost, it loses 256 characters at a time. The Editor will salvage the rest of the file, but you should be careful at all times to keep the diskettes and disk drives clean and protected from contamination.

### SALVAGING TEXT FROM A BAD DISK

When the Editor is reading a file and encounters an unrecoverable disk error, it displays a message, waits for you to press "ENTER", and then attempts to read the rest of the file. The unreadable section(s) of text are placed in memory by square graphics. Normally, these sections are multiples of 256 characters in length, but the degree of success in error recovery is always uncertain. When this kind of error occurs, examine the text, replace the marked area(s) by what ever you think used to be there, and then save the file onto a different disk. A lost sector of text isn't serious: since it only fills three or four lines on the screen, you usually will be able to figure out what had been there and re-create it.

Sometimes, reading the disk again or using a different disk drive will succeed, but even if it does, the diskette should be considered suspect. When you find you have a bad disk, copy what you can from it to another disk, and put the disk away, never to be used again unless you need to try to copy a file from it later on. No matter how fast you type, it takes hours or days to fill a disk that cost less than \$3.00. Is the value of your time low enough to risk having to re-do a day of work just to save \$3.00?

### HOW MESSAGES ARE NUMBERED

All error messages in ALLWRITE are numbered, so if the brief explanation that is displayed isn't sufficient, you can refer to this chapter for details. If ALLWRITE is reporting a DOS error, you may have to check your DOS manual for details, since the error numbers sometimes vary from one DOS to another.

Error numbers below 200 are DOS errors; numbers from 200 to 230 are Text Formatter errors; and numbers from 231 to 255 are Editor errors. Not all numbers are in use.

**DOS ERRORS**

These messages can occur anyplace in ALLWRITE. Any error number below 200 not listed here should be found in your DOS manual. When ALLWRITE displays one of these errors, the message will just say "Disk Read/Write Error."

**1 - 7, 9 - 14, 17 - 18, 20 - 23, 42 - 199. Bad Disk**

If any of these errors occurs, the diskette is bad, hasn't been formatted, is in the wrong density, or was created by an incompatible operating system. If you know the disk used to work, try reading it a couple more times, or try it in a different disk drive. If you have a head cleaning kit, you can try cleaning the head. If you have a disk drive diagnostic program, you can see if your equipment is at fault. However, most of the time, the problem will be with the diskette itself.

Action: Copy the good files to another disk, one at a time, by groups, or with "TRANSFER". Don't bother trying to "BACKUP" the disk, because the physical error will cancel the entire backup operation. ALLWRITE's Editor will salvage the readable sectors from any text file on the disk.

**8. Drive not available**

Action: close the door, turn the drive on, insert a disk, make sure the cable is connected. If none of these work, even after pressing the "RESET" key, you may have a bad disk drive. Confirm this by trying to display the Directory from DOS, without using ALLWRITE.

**15. Write protected disk**

If you really want to write something on the diskette, remove the write protect tab. If that doesn't work, you have a bad disk drive or a diskette whose write protect notch doesn't line up with the write-protect sensor. See if other disks work.

**19. Improper file I.D.**

Retype the file name properly. We cannot give rules for this here, because the operating systems differ in what they allow. If you get this message right after switching from one part of ALLWRITE to another, it may mean the file name was passed along incorrectly. In that case, just type it back in.

Some DOS's do not accept lower-case commands, so after exiting from ALLWRITE, just press SHIFT ZERO to restore upper-case.

**24. File not found**

The file isn't on any on-line disk. Make sure you typed the name properly. If so, check the directory of the disk to make sure the file is there, and if not, insert the correct disk and try again. Also see Error 41.

**25. Access denied due to password protection**

Include the correct password when you retype the name. If there was no password on the file, a disk or DOS error may have altered the directory of the disk incorrectly. This can be corrected if you know how to use a "ZAP" program to modify the Directory, but if you don't know how to do this, experiment on a backup disk or get help from someone who knows what to do. PROSOFT does not offer help with this sort of thing, and we don't use passwords ourselves.

**26 - 27, 30. Disk full**

Use another disk. Some DOS's give this message when the disk is write-protected, even if it isn't full. If the message is #27, you should "Kill" or "Remove" the incomplete file from the disk to avoid problems later on.

**28 - 29. Attempted to read past EOF or past file limits**

Bad directory entry, diskette incompatible with your DOS, or ALLWRITE programming error. If the file can be listed to the screen by DOS, but not read by ALLWRITE, report the problem to us as explained in Chapter one. You might re-install ALLWRITE, a glitch could have developed.

**31. Program not found**

Same as #24

**32. Improper Drive number**

You specified a drive number higher than the maximum allowed in your configuration. "3" or "7" is usually the limit.

**34. Attempted to use non-program file as a program**

A data file was used as a command

**35, 36 Memory fault during program load**

This can mean one of two things: 1) the program is not set up for use with your computer, or 2) part of the RAM (memory) in your computer is not working properly. The second possibility is not uncommon with TRS-80 Model I's. If you try to use a Model 4 version of ALLWRITE in Model III mode, you will get this message.

**37. Illegal access attempted to protected file**

Same as #25

**38. I/O attempted to unopen file**

Probably an error in ALLWRITE. Contact us. (or re-install)

**39. Device in use**

Probably an error in DOS, or an equipment error.

**41. Device not available**

If you are using Model 4 TRSDOS 6.0 or 6.1, this means a file was left "Open" when the computer was RESET. The solution is to 'RESET' the file itself and the DOS command is "RESET fileid". If you're using any other DOS, the message means what it says: see message #8.

**ALK (KEYBOARD DRIVER ERROR MESSAGE)****Please install ALLWRITE first.**

ALK cannot be used until after the installation program "ALINSTAL" has been used. This is necessary because keyboard support must know which DOS you are using.

**EDITOR ERROR MESSAGES****PLEASE ENTER DOCUMENT I.D. (default = xxx)**

When the Editor starts or restarts, it displays this message unless a file name was passed to it as a parameter or by another program. The "default" portion is shown only if a file name is already known to it.

**This would be a new file**

If the file name is not found on any available disk, this message is given as a warning. If the file really is new, just press "ENTER" to start writing to it. Otherwise, insert the correct disk or type a different name.

**PRESS ENTER TO CONTINUE**

This isn't an error message. It's issued by many features in the Editor, usually after some other message has been displayed. After reading the other message and deciding what you want to do next, press "ENTER" to return to normal editing.

If you press "CLEAR" instead of "ENTER", the Editor will usually display the "CONTROL" cursor on the text screen. This can save you a keystroke if the next thing you wanted to do is use a control key.

If you press the asterisk '\*' instead of "ENTER", the Editor will usually do a "screen print" before re-displaying the text. The "CLEAR" key should not be pressed along with the asterisk.

**243. Memory not available**

Displayed when the "AREA" command is issued on a Model I or III, or when the second or third bank of memory is unavailable on a 128K Model 4. It means you cannot use the extra memory because it is not installed on your computer or is being used for something else (spooler or MEMDISK).

**244. No such link in file**

The "LINK FORWARD" command was issued, but the current file in memory doesn't end with the ";AP" control word; or the "LINK BACK" command was issued, but the current file doesn't begin with the ";BA" control word. If you want to use chained files, you must link them together with these control words, as explained in Chapter 4, under "Long Documents." ";BA" and ";AP".

**245. Target can't be inside block**

A block cannot be moved or copied to a point within itself. This message means that <H> was pressed while the cursor was between the <B> and <M> or <C> points that defined the block. The text is not affected, and block mode is cancelled. You can try again, positioning the cursor correctly.

**246. No text in memory**

You tried to SAVE text or perform some operation such as block move, SEARCH, etc., with an empty document. No harm is done, but your request is ignored.

**247. (unassigned)****248. Text not found**

A "SEARCH" was requested, but nothing was found to match it in the remainder of the file; or a "REPLACE" was requested, but the cursor was not positioned at text that matched the current "Search argument". This isn't necessarily an error, but an informational message.

**249. Missing or invalid parameter**

The value(s) following a command were incorrect or missing. Certain commands must be followed by parameters (words or numbers), and if those parameters are missing or do not match the allowed values, this message will be given. No harm is done, text is not changed, and the command is ignored. You can use the "?" command to see what you just typed, and then correct it and press "ENTER" to try it again.

**250. Unknown command**

The first thing you typed on the command line didn't match any of the commands known to the Editor. Certain commands require 2 letters for safety or clarity, so if you typed "E" or "Q" for example, they would be rejected with this message. "EN" or "QU" are required in these examples.

**251. No more room in memory**

This can be caused by several things, but it usually means you've typed so much that memory filled up. In this case, the Editor may not be able to store the last few lines you typed. To solve the problem:

1. save the file to disk
2. Block Put some of it to another file;
3. Block Delete the "Put" block from the original;
4. ";AP" (append) the second file to the first;
5. save the first file;
6. continue your typing in the second file.

A block copy may find there is not enough memory to duplicate the block, in which case no duplication will occur.

A Search-and-Replace or Change may run out of memory if the Replacement argument is longer than the search argument. The replacements are made until memory becomes full, so text in memory has been changed. No text will be lost, but not all the replacements will have been made.

"GETFILE" may not have enough room to include all the relevant text from the second file. As much text as possible will be read, and the rest of the second file will not be included into memory.

Reading a large primary text file into memory can cause this error also. If it happens, the end of the file will not be in memory. There are a couple of ways to solve this problem. If you're using our "ALK" keyboard driver, just exit from ALLWRITE, turn off the keyboard driver ("ALKOFF") which will give you some more room in memory, and use "AL" without

"ALK" to edit the file again. This time, it will fit and you can split it into two smaller files using "PUTFILE". You can insert the ";AP" and ";BA" links before doing this, which will save time and reduce the likelihood of errors later on. After "Putting" the second half of the file to disk, you can delete its text from the first file, then save the new, shorter first file. Finally, you can start "ALK" again, and continue editing each of the two file segments separately.

If you aren't using "ALK", you can use the "NAME" command to change the file name immediately, then to delete a little more text from what did fit in memory, add an ";AP" (append) at the end of the file, save it to disk under its new name (be sure to change the name first, or text will be lost). Write down the last few words of text in this modified file. Then, use the "NEW" command to start the Editor over again, and specify another new name (not any of the names you've been using). Next, use "GETFILE" to read the original file (the one that was too big), but specify those "last few words" as the starting point for a selective "GET". Only the end of the file will be read, and there should be enough room for it. "GETFILE" reads text at full speed, so there are no drawbacks to this method. You must be careful about the filenames when using this approach, or you can lose the original file by accident.

## 252-255 (unassigned)

### TEXT FORMATTER ERROR MESSAGES

When the Text Formatter detects an error in a control word, it prints or displays one of the messages below, prints or displays the line containing the control word, underlines the error, and switches to "Summary mode". If the ";ER" control word specified a different action than this default, that other action will be taken instead. If "E" (mini-edit) was specified, then the error is not printed on the paper, so you can try to correct it and keep going.

The Formatter also issues prompts for various reasons. These are not errors, and they are listed after the error messages.

#### **200. Unknown command**

A control word not supported by ALLWRITE was used.

#### **201,202. Number too high or low**

One of the numbers used as a control word parameter was incorrect.

#### **203. Parameter missing**

A required parameter (number or word) after a control word was not found.

**204. Margins conflict**

The top or bottom margins are too small to contain the number of titles you've specified; the top and bottom margins are bigger than the page length; the indents and/or offsets and/or hanging indents are bigger than the line length

**205. Line too long**

A control word line cannot be longer than 127 characters.

**206-207. (unassigned)****208. Undefined table**

A ";TR" or ";PI" control word specified a table that had not been read into memory by an ";XT" or ";WT" control word at the beginning of the text file. Either identify the correct table or add the ;"XT" or ;"WT" at the very top of the document. This applies only to certain daisywheel printers.

**209. Command not allowed here**

";XT" or ";WT" was used too late in the document, or ";FN" or ";FT" was used before an ";FF" was found. Please see chapter 5 for details regarding the proper use of these control words.

**210. Bad column definition**

The number of columns must be between 1 and 16, and the total width of the columns cannot exceed the line length. If your line length is '80', you cannot specify ";CB 4,25", since that would require a line length of '100', not counting gutters between the columns.

**211. Too many imbeds**

An imbedded file can imbed another file, but the limit on this sort of thing is "4", including the imbed issued in the primary file. This means at most five files can be involved at a time. The extra imbeds are ignored.

**212. Parameter invalid**

A number or word following a control word cannot be used with that control word, or is in the wrong place. If the error isn't apparent when you examine the text, check the control word's use in chapter 5.

**213. Label not found**

A ;"GO" or selective ;"AP" or ;"IM" referenced a label, but no ;"LB" bearing that label was found in the remainder of the file or the appended / imbedded file.

**Text Formatter Prompts****Enter Filename (Default = 'xxx')**

This means the Formatter is starting a new file in prompt mode (<">), but a control word, such as ;IM or ;RD, specifies a filename that is invalid or not found. Put the proper disk in the correct drive and type in the correct filename, or press <ENTER> if the default is correct. Typing "?mask?" will display a directory in the same manner as the editor's masked directory. Typing ";" will activate Mini-edit, if the prompt is due to the Formatter starting, rather than to an error. Typing "-" will cause the Formatter to go to its exit menu.

**xxx is a bad filename. Enter filename (default = 'xxx')**

A filename does not follow the proper syntax for a filename. Type in the new filename as described in the previous message.

**xxx not found. Enter filename (default = 'xxx')**

A file is not on any of the disks currently available to the computer. Put the correct disk in a drive and follow the procedure described previously.

**Enter options**

This is asked when a file is to be printed due to <">. It allows you to type in a single line of control words, complete with semi-colons and ending with "ENTER". If you don't want any options, just press "ENTER".

**Do you want to continue processing (Y/N/E, default=Y)**

This is displayed when you press "BREAK" during printing or previewing. If you reply "Y" or just press "ENTER", processing continues as though nothing had happened. If you reply "N", printing and processing stop immediately. If you reply "E", then "Mini-edit" is activated. A few more lines of text may be processed before Mini-edit begins to function.

**Eject page (Y/N/C, default=Y)**

This is displayed when the Text Formatter finishes a document unless it was running in automatic mode (Editor's Soft Key Shift Capital 1). If you reply "Y" or just press "ENTER", the last page is ejected from the printer. If you reply "N", the page is not ejected. If you reply "C" ("continue"), Mini-edit is activated, and you will be prompted to "\* ADD" more text, which will be processed as part of the file and printed next. If you do this, you can stop by just pressing "ENTER" without any text.

**Do you want to print the table of contents (Y/N, default=Y)**

";TC" entries were processed in the document. Reply "Y" or press "ENTER" to print the T/C; reply "N" if you don't want it printed. The table of contents file will be left on disk either way.

**Do you want to print the index (Y/N, default=Y)**

";IX" entries were processed in the document. Reply "Y" or press "ENTER" to run the index generation program and have the index printed, or "N" if you don't want it processed. The index file will be left on disk either way.

**Press P to print again, E to edit, Q to quit or T for top of next page (default=P)**

This is the last message from the exit menu. "E" runs the Editor again, "Q" exits to DOS, and 'P' or "ENTER" restarts the Text Formatter.

**\* ADD \***

This is one of the mini-edit prompts. You may type in a line to be processed just as though it came from a normal disk file. If you are finished, just press <ENTER>.

**Press <ENTER> to confirm EOJ, or type more text**

When a blank line is typed in mini-edit '\* ADD \*' mode, it means that the file is finished; if you did not mean to finish, type more text; if you did mean to finish, confirm by pressing <ENTER> a second time.

**\* EDIT \***

This is the prompt for the other form of mini-edit; it lets you modify a file as it is read from disk.

**Use T,I,R,E. or D**

This is an error message issued by the "\* EDIT \*" form of mini-edit when a command other than T,I,R,E, or D is used.

**Enter substitution value # 0 (date)**

The special variable, '@@ 0' was encountered for the first time in the text. Reply with whatever value you want to use in place of it, or just press "ENTER" to set it to nothing.

**Enter selection code**

An ";RD" control word was encountered for the first time, and included a selection symbol such as the at-sign "@". If you want names and addresses to be extracted selectively from the list, give the selection code as the reply. If you want to use the entire list, just press "ENTER".

**Press <ENTER> when ready**

This is displayed when a ";ST" control word is encountered. The text Formatter stops reading from disk until "ENTER" is pressed, but continues to print until its internal buffer is empty. This message is also displayed after most of the remaining messages in this chapter.

**Insert new page**

Cut sheets are being used, and another piece of paper is needed.

**Pause encountered in text**

The emphasis mark "@ >" was encountered. Change print wheels, or whatever you need to do, then press "ENTER".

**Change wheel to pitch xxx, sequence yyy**

A ";TR" or ";PI" control word was encountered and ";WC Y" is in effect. After making the change, press "ENTER" to continue. If you are printing a rough-draft and only a pitch change is involved, you can press "ENTER" without bothering to change the print wheel first.

**Page range finished, Enter new range, or <ENTER> if finished**

The page range specified by ";PG" has been finished. If you would like to print yet another range, specify it at this time; otherwise, press <ENTER> to indicate that you are finished.

**Disk I/O Error # xxx in file yyy**

An error has occurred while reading a disk file. The error number is the DOS error number. The Formatter cannot properly format the rest of the file, and so it goes to its exit menu.

**Out of memory**

The Text Formatter cannot continue processing once this happens, so it goes to its exit routine. Insufficient memory can occur when you use multiple columns (particularly with video preview), footnotes, or variables. To correct the error, you must reduce the number of characters per page (columns), limit the size of the footnotes that must be printed on the page, or use fewer / shorter variables. This error will not occur in "reasonable" situations. On Models I and III, there may not be enough memory for ";VI" (video) previewing of multiple columns.

**Index too large**

The Index processor, "ALINDEX/CMD" displays this message if it runs out of memory. You may be able to free some extra memory by re-booting the computer and running ALINDEX as a DOS command, without using the ALK keyboard driver. Otherwise, you will have to remove some index references from your text and then re-print the document to generate a smaller index.

## **CHAPTER 8**

### **SUMMARY OF FEATURES**

This chapter lists the features of the Editor and the Text Formatter. It is intended for quick reference by people already familiar with word processors or ALLWRITE, and does not replace the rest of this book. If the summaries in this chapter are insufficient, please refer to the rest of the book for detailed explanations. No attempt has been made in this chapter to show options available on advanced features.

Page references for further information are indicated by  symbols. The text Formatter is summarized twice: the first list is alphabetical and indicates whether or not a control word causes a "control break". The second list is topical, and corresponds to the Soft Key Cue Card. The second list has slightly more information about command functions.

#### **EDITOR**

The Editor is used to type in and change text. Editor functions are divided into five categories: typing, arrows, control keys, soft keys, and commands. To use a control key, press 'CLEAR' first. To cancel control mode, press 'CLEAR' again. To use a command, press "BREAK" first, and "ENTER" afterwards. To cancel command mode, press "BREAK" instead of "ENTER". Soft keys are used like control keys.

#### ARROWS §52 - 54

Left	move cursor one position to left. If at left edge, move to right edge of previous line
right	move cursor 1 position to right, if at right edge, move to start of next line
up	move cursor one line up, if at top of file, open a blank line
down	move cursor 1 line down, stops before last line moves off screen
SHIFT	left move cursor to left side of line - if at left, move up one line
SHIFT	right move cursor past end of text on line - if at right, move to next line
SHIFT	up move cursor to top of screen - if at top scroll back 1 screen
SHIFT	down move cursor to bottom of screen - if at bottom scroll forward 1 screen

#### **Editor Control Keys**

B	define start of a block §84
C	if after <B>, define end of block to be copied; if by itself, define one line to be copied to <H> §86
D	1) if not in block mode, delete character at cursor §55 2) if in block mode, define end of block to delete §87
E	scroll to end of text file §58
F	repeat previously-defined "Find" (Search) §96
G	display/remove column/tab grid §106
H	move/copy currently-defined block "H"ere §86
I	insert mode on/off §57
J	join lines together, save screen §115
K	key replicate: duplicate character before cursor §119
L	1) if not in block mode, delete rest of screen line §56 – 2) in block mode, define end of block to list (print) §87
M	if after <B>, define end of block to be moved; §86 if by itself, define 1 line to be moved to <H>
N	open a new, blank line below current one §114

O one-key insert §87  
 P 1) if not in block mode, delete rest of paragraph §56;  
 2) in block mode, define block to be put to disk §88  
 Q end a Soft Key definition §146  
 R repeat previously-defined "Replace" §98  
 S delete rest of sentence §56  
 T scroll to top of text file §58  
 U scroll line to top of screen (half-screen scroll) §119  
 W delete rest of word §56  
 : define 3-digit ASCII value §117  
 = hard space symbol, ASCII 127 §117  
 - underline on screen §116  
 / repeat previously-defined "Search", same as "F" §96  
 > reverse word at cursor with next word §116  
 \* screen print. When "PRESS ENTER TO CONTINUE" §113  
 is on screen, use "\*" without CONTROL  
 @ tab to next word §105

right arrow tab right §105  
 left arrow tab left §105  
 SPACE delete rest of line, same as "L" §56  
 ENTER split line at cursor or open a blank line §114  
 Comma define a point: press any letter for I.D. §103  
 Period recall a point: press any defined letter §103

to set tabs: display grid, move cursor onto grid §106  
 plus sign sets tabs, minus sign clears them §107

### Editor Soft Keys

These are the distributed defaults:

1 new paragraph §180  
 2 new page §191  
 3 center next line §178  
 4 underline on §160  
 5 underline off §160  
 6 boldface level 2 on §160  
 7 boldface off §160  
 8 validate emphasis marks §123  
 9 display file status §59  
 0 HELP §65

! run Text Formatter automatically, allow no options §74  
 " run Text Formatter, allow options §75  
 # run DOTWRITER (if you own it) §44  
 \$ run ELECTRIC WEBSTER (if you own it) §43  
 % link back to previous ";BA" file §134  
 & link forward to next ";AP" file §134  
 ) exit to DOS §65

Editor Commands

Area	1, 2, 3	switch memory areas (128K Model 4 & 4P) §130
Autosave	n	set new autosave value §122
Case	Y,N	honor/ignore U/L case in SEARCH §102
Change	/x/y/n	change 'x' to 'y' 'n' times §100
Dir	n	display Directory for drive 'n' §64
End	fid	save file and restart/exit Editor §127
Find	text	define SEARCH and look for it §93
Flow	Y,N	Y="ENTER" same as blank in SEARCH §103
Getfile	fid	get (read) all or part of a file §89
Help		display on-line HELP menu §63
Join	Y,N	NO = add "ENTER" to end of each line §120
Key	c,L,S,fid	define, load or save soft keys §145
Kill	fid	kill/remove a file from disk §126
Length	n	sets screen line length §121
Link	B,F	link back or forward a file §136
Locate	text	define SEARCH and look for it §93
Lup	text	SEARCH backwards (up) §95
Name	fid	change file I.D. of text in memory §126
New		restart Editor if no changes in memory §66
Options	P,R	set 'prompt' or 'run' option §129
Printer	1, 2, 3	select current printer §122
Quit		exit to DOS without saving changes §65
Replace	text	define REPLACE and do it if match §97
Run	fid/CMD	exit ALLWRITE, run another program §128
Save	fid	save text to disk §62, 125
Search	text	define SEARCH and look for it §93
SR	n	repeat SEARCH and REPLACE 'n' times §99
Status		display file statistics §59
Sup	text	SEARCH backwards (up) §95
Tabfile	S,L,fid	save or load a tabfile §108
Validate	sym	check for unbalanced Emphasis Marks §123
Whoops		cancel recent minor screen changes §61
X,Y	n	execute soft key X or Y 'n' times §149
Z	n	execute soft keys X and Y 'n' times §149
?		display last command §123
=		repeat last command §123

TEXT FORMATTER

The Text Formatter is used to print text according to specifications you provide and defaults that are taken when you have not made a specification. There are two kinds of Text Formatter specifications: Emphasis marks, which begin with an at-sign "@" and may appear within normal text; and Control words, which begin with a semi-colon and must be on lines of their own. The semi-colon must be preceded by an "ENTER" symbol on the previous line, and the control word or its parameters must be followed by another "ENTER" symbol. In some cases, more than one control word may appear on the same screen line.

**Formatter Emphasis Marks §160**

@\$	underline on	@%	underline off
@*n	boldface level 'n'	@*0	boldface off
	'n' may be 1-9		
@(	double width ON	@)	double width OFF
@/	italics ON (red)	@?	italics OFF (black)
@+	start super-script	@-	start sub-script
@=	cancel sub and super-script on EPSON, MICROLINE, etc. on other printers, sub and superscripts cancel each other		
@<	backspace (overstrike)		
@>	pause printer		

**Formatter Control Words, Alphabetically**

This list is arranged alphabetically. It indicates whether or not the control word causes a "control break", and whether it must be the last or the only one on the screen line. The second list is arranged by topic and summarizes the function of each control word, semi-colons are not shown. Page references in "§".

**Key:**

NO BREAK	text on next line may be connected to text on preceding line.
LAST	must be last control word on line
ONLY	must be only control word on line

When these keys are not shown, the control word: 1) causes a break in the text; 2) may appear anywhere on the line, and 3) may be followed by other control words. Remember that a control word line must begin with a semi-colon and end with an "ENTER" symbol, and be at most 127 characters long.

CW PARAMETERS	MEANING	RESTRICTIONS
AP fid,label	append to 'fid' §134	NO BREAK, ONLY
AR var op var	arithmetic §271	NO BREAK, LAST
BA	treated as comment §134	ONLY
BM 6	bottom margin §170	
BR	force control break §190	
BT	bottom title §203	
CB n,w,U	multiple columns §222	
CC n	carbon copies §221	NO BREAK
CE n,ON,OFF	center §178	
CM text	comment §219	NO BREAK, LAST
CO ON,OFF	concatenation §189	
CP n	conditional new page §193	
CS Y,N	cut sheet pause §197	
CW symbol	control word symbol §183	
DA n	boldface value §218	NO BREAK
EB	even page bottom title §203	
ED	activate Mini-edit §165	NO BREAK
EM 5	even page left margin §171	
EN label	end of a label block §142	NO BREAK
ER S,E, Q	what to do on an error § 224	
ES symbol	emphasis mark symbol §183	

<u>CW PARAMETERS</u>	<u>MEANING</u>	<u>RESTRICTIONS</u>
ET	even page top title §198	
FF	activate footnoting §237	
FL	read small file to memory §239	LAST
FN	start or end a footnote §261	NO BREAK, LAST
FO ON, OFF, RIGHT	format control §187	
FT	footnote title §257	
GO label	skip to 'label' §143	NO BREAK, LAST
HE	simple heading §198	
HI	automatic hanging indent §176	
HY c, ON, OFF	hyphenation §173	
IF condition	skip to next IF' if FALSE §272	NO BREAK, ONLY
IM fid, label	imbed text §138	NO BREAK, LAST
IN	left margin indent §175	
IX text	index entry §250	LAST
JU	justification control §189	
KE n	keyboard entry §220	NO BREAK, LAST
LB label	defines a label §142	NO BREAK, ONLY
LG n	logo space margin §172	
LI 6,7,8,12	lines per inch §185	
LL 65	line length §171	
LM 5	left margin §170	
LS 1,1.5,2,3	line spacing §184	
NA ON, OFF, A, I, R	print file names §225	NO BREAK
NO AP, IM	ignore certain control words §217	
NU 8,2	legal line numbering §244	NO BREAK
OB	odd page bottom title §203	
OF	non-automatic hanging indent §176	
OM 5	odd page left margin §171	
OT	odd page top title §203	
PA n, O, E, D	new page §191	
PG n1, n2	page range §194	
PI 0,1,2	proportional pitches §206	NO BREAK
PL 66	page length §186	
PN ON, OFF	suppresses page numbering §194	
PP 5,1	paragraph §180	
PS \$, char	page number symbol §200	
PT n	point numbering §182	
RD c, fid	form letters §227	LAST
SD n, Y, N	micro-spacing §208	
SE var val	set variable value §269	NO BREAK
SK n	skip 'n' blank lines §179	
SM	summary mode §78	
ST message	stop for operator §221	NO BREAK, LAST
SY n1, n2	print special symbols §267	NO BREAK
SY @n1, n2..	special escape sequences §267	NO BREAK
SY /	define 1 graphics line §267	BREAK, ONLY
n1, n2.../	ASCII graphic codes for SY/	
TB c, t1, t2, ...	define tabs §109	
TC text	table of contents entry §248	LAST
TM 6	top margin §170	
TR @	use internal xlate table §264	NO BREAK
TR c	do character translation §264	NO BREAK
TR n1, n2, n3	translation range §265	NO BREAK

<u>CW PARAMETERS</u>	<u>MEANING</u>	<u>RESTRICTIONS</u>
TT n/l/c/r/	top title line §198	
US text	underline §219	NO BREAK, LAST
VI	non-stop video preview §76	
VS	screen-at-a-time video preview §77	
WC Y,N	pause for wheel change §216	
WD ON,OFF	anti-widowing §192	
WT c,fid	identify width table §214	ONLY
XT c,fid	identify translate table §213	ONLY

### Formatter Control Words by Topic

Semi-colons not shown. Grouped by function. Defaults shown where possible. Number and words in lower case should be replaced by your specifications.

#### Margins §169

TM 6	top margin in lines (6=one inch)
BM 6	bottom margin in lines
LM 5	left margin in tenths of an inch
LL 65	line length in tenths of an inch
PL 66	page length in lines (66=11 inch paper)
OM 5	odd page left margin
EM 5	even page left margin
LG n	logo space margin for first page only

#### Indentation §174

PP 5,1	paragraph indent, blank lines
IN n	left margin indent in tenths of an inch
CE n, ON, OFF	center next lines
HI	automatic hanging indent setting
OF	non-automatic hanging indent
TB c, t1, t2,...	define tab symbol and stops modifiers: D (decimal, R (right), C (center)

#### Spacing §184

SK -n	leave 'n' blank lines unless at top of page
PA n, O, E, D	new page: number, ODD, EVEN, DELAY
CP n	conditional new page
LI 6, 7,8, 12	lines per inch
LS 1, 1.5, 2, 3	line spacing (single, double, etc)
WD ON, OFF	anti-widowing

Text Justification §187

FO ON, OFF, LEFT RIGHT, ALL	ON = left and right justify text except last line of paragraph
JU	like "FO" but does not override "CO"
CO ON, OFF	concatenation (join lines together)
BR	break; starts a new print line

Pitch §204

PI 0, 1, 2 8,10,12,16 3-6,20,21 9,11,13,17 A-Z	proportional pitches, printer dependant §206
SD n, Y, N	monospace pitches, printer dependant
	special purpose, some printers
	better quality monospace, some printers
	use a pitch defined by ";WT" or ";DF"
	micro-space definition controls §208

Width and Translate Tables §210

XT c, fid	use translate table in file 'fid' §214
WT c, fid	call it letter 'c'. Referenced via ";TR"
DF c, fid	use width table in file 'fid', §213
TR c	call it letter 'c'. Referenced via ";PI"
TR @	use font and width tables of "fid"
TR n1,n2,n3	call it letter "c" referenced via ";PI"
SY n1,n2	do character translation using table 'c' §264
SY @n1,n2	use internal translation table for printer
SY /	define range of translations
n1,n2,../	special symbols, count with text §267
	escape sequences, not counted
	generate one print line of graphics
	ASCII codes for ;SY/

Pausing

CS Y, N	pause for cut sheet paper change §197
ST message	stop reading disk, wait for "ENTER" §221
WC Y, N	pause for wheel changes §216
KE n message	keyboard entry 'n' lines §220

Files

AP fid, label	append (link) to 'fid' §134
IM fid, label	label is optional starting point in 'fid'
RD c, fid	imbed text in 'fid' §138
FL	label is optional starting point in 'fid'
BA fid	form letters, mailing labels §226
TC text	store form letter in memory §239
IX text	treated as comment §134
NA ON, OFF, A, I, R	table of contents entry §248
	index entry §250
	print file names during Append, Imbed §225

**Titles and Headings** §198

TT n/l/c/r	top title line 'n'; three segments
BT	bottom title, similar format to TT
OT	odd page top title; similar format
ET	even page top title; similar format
OB	odd page bottom title; similar format
EB	even page bottom title; similar format
FT	footnote title, similar format
HE	simple heading, same as TT/text//
PS \$, char	page number symbol in titles §200
PN ON, OFF	suppress page numbering §194

**Numbering**

PT n	point (bullet/paragraph) numbering §182
NU 8,2	legal line numbering §244
AR var op var	arithmetic §271
SE var val	set value into a variable §269

**Logic**

IF condition	skip to next 'IF' if FALSE §272
GO label	skip to 'label' later in file §143
LB label	defines a label for GO, AP, IM §142
EN label	end of a LABEL block for IMBED §142

**Preview** §76

VI	preview to video display, run continuously §76
VS	preview to video, stop each screenfull §77
SM	summary mode: print 1st line of each page §78
ER S, E, Q	on error: start summary, mini-edit, Quit §224

**Footnotes** §255

FF	define footnote format, activate footnoting
FT	define footnote title(s)
FN	start, end, or retrieve footnotes

**Multiple Columns** §222

CB n, w, U	'n' columns, each 'w' tenths of an inch wide if no values, start a new column immediately. 'U' forces unbalanced column
------------	---

**Overrides**

ED activate Mini-Edit mode §165  
NO AP, IM ignore these (and other) control words §217

**Miscellaneous**

PG n1,n2 page range; only print what's in it §194  
HY c, ON, OFF hyphenation §173  
DA n boldface value, added to emphasis mark §218  
US text underline, but not between words §219  
CW symbol select char to replace ";" on ctl words §183  
ES symbol select char to replace "@" in emphasis §183  
CM text comment, ignored thru next "ENTER" §219  
CC n carbon copies (print 'n' copies) §221



## CHAPTER 9

### WRITING WITH A WORD PROCESSOR

There are two primary reasons for using a Word Processor. The first one is pretty obvious: It will save you time, and that reason, all by itself, will more than justify the initial investment in money and learning. The second reason, However, is even more important: a Word Processor can change the way you write, and thereby improve the quality of your writing, not just the quantity. This is due partly to the incredible flexibility a Word Processor gives, partly to the tremendous amount of time that is saved, and partly to the avoidance of boring, repetitive, mechanical tasks such as re-typing a page because two words have been changed.

Let me give a simple example of this. Suppose you've written a 10-page paper (a sales proposal, a term paper, or a magazine article), and now it's Sunday night, and the paper is due first thing Monday morning. (I realize this is unrealistic, since no one except me would wait until the last minute to complete something important. However, please humor me a moment longer.) As you're proof-reading the final copy, you find, to your horror, that you've omitted a key sentence from the introduction. What will you do? (What will you do?) If you hand in the paper as-is, a major part of its impact will be lost. To add the sentence, you'll have to re-type the entire paper, since everything after that introduction will have to move down three lines, and then onto the next page, etc. At your normal typing speed, this will take several hours. If you're in business, and you think you can just give it to your secretary in the morning, you might have to arrange for him/her to be in a couple hours early.

If the paper was prepared on a typewriter, you've got a real problem on your hands. However, if it was done on a Word Processor, there's no problem at all: you can just put the diskette back in the computer and make the change in less than a minute. Then it'll take 5-15 minutes to reprint the entire 10 pages, depending on the speed of your printer. During that time, you can do something else.

This example only showed how you can save time by using a Word Processor. Now, let's extend the idea a bit. If you have to do some research before writing a paper, and you have a month in which to complete the project, you'll probably plan to divide your time between researching, analyzing, outlining, writing, and revising. Having gone through this many times until the late 1960's (I was lucky enough to start using a good Word Processor in 1970, and haven't done things the old way since), I'd estimate that at least a third of the time would have been to be set aside for the outlining, writing, and revision steps. I'd also be willing to bet that the last few days of the month would be spent under considerable pressure (arterial as well as psychological and financial), with frequent vows to allow more time on the next project.

However, if the paper could be done with a Word Processor, a lot more time could be spent in researching and analyzing the results of that research, since you'd be secure in the knowledge that the actual writing and cleaning up will take very little time. Consequently, you'd be able to produce a higher quality result, possibly discovering some things you otherwise would have missed.

By the way, if you're thinking that your own writing is limited to memos and letters, so that none of this applies to you, please read on: I'll talk about letters later on.

### Writing With a Word Processor

(some of this section is based on material written by Bruce Powell Douglass of A Priori Software and originally published in "NEWSRIPT 7.0" in 1982.)

To write something of any length or complexity requires planning, and one of the best ways to plan is through the use of an outline. If an outline is written on paper, it quickly will show erasures, cross-outs, and arrows. This is a time-honored way of developing one's thoughts, ensuring that all important topics will be covered, and then arranging those topics in the most effective order. For example, the outline of this chapter was:

1. purpose - what should this paper accomplish?
2. introduction - how paper came to be written
3. reasons for using a W.P. - give examples
4. writing with a W.P. - follow BPD's approach
5. using a W.P. for form letters
6. spelling checkers
7. summary

Actually, the original outline was not nearly this orderly and had it been on paper, the erasures, etc. would show. Since it was done with a Word Processor, the changes and improvements were made with casual ease, and rather than wasting time, they merely reflected the evolution of my thinking. Not only was there a time savings, but also a better final result. Example: after finishing one draft, I decided to mention Spelling Checkers. All I had to do was write these additional sentences, add a new item #6 to the outline, and then write the relevant paragraphs. There was no hesitation in making and implementing this decision. Why should there have been? After all, it didn't mean I'd have to retype anything, let alone the whole paper!

If this outline had been on paper, the next step would have been to take a fresh sheet of paper, copy the first item ("introduction"), and either fill in some additional details, or else start writing what I thought belonged in the introduction. The point here is that it would have taken some time to re-copy each thing from the first step of the outline to the next step, and during the time needed to make that mechanical transfer; I could not have been giving full concentration to what I wanted to say in the introduction. I'd have been paying so much attention to the mechanics, and trying so hard to avoid making mistakes, that I'd likely have lost my train of thought on complex or subtle issues.

Since the outline was on the screen of my computer, and controlled by a Word Processor, all I had to do was hit a couple of keys, open up some space between the lines, and fill in additional detail. Moreover, whenever it became clear that material belonged elsewhere, it took only a few seconds to tell the Word Processor to move those lines to where it seemed they should go. That made it simple for me to re-read the material in its new context, and then to decide whether it really was better off in the new location. If not, it could be moved again, deleted, or modified, until I was satisfied. No effort, just results.

Because all this took only a few seconds, I never hesitated to make these kinds of changes. However, if I'd been writing with pencil and paper, the sheer effort of doing cut and paste would have limited my ambition, and I would have settled for whatever could be done in the time available.

Another, unanticipated method of writing emerges when using a Word Processor: the ability to skip around without fear of losing one's train of thought. When using paper, writing tends to be done from the beginning to the end. If something has to be added in a completed section, the tendency is to make a note for later on, and then come back to it, trying to remember just what the brilliant idea was. The alternative is to end up with a large pile of small pieces of paper. On the other hand, when using a Word Processor, it takes only a couple of keystrokes to move back, make a change, and then resume. Again, not only is there a substantial time savings, but there also is an improvement in quality.

### **Form Letters**

A lot of what goes into a letter is repetitive. The extreme case is the situation where the same Form letter has to be sent to several (thousand) people. In an attempt to personalize such letters, computers can be used to place the recipient's name and address in the salutation, and possibly even in the body of the letter. (Suggestion: turn off right justification and use a solid-character printer to make the result look typed, not computer generated.) However, there may be many other kinds of letters, and they all can benefit from Word Processing.

The closing of a letter is very standard:

Sincerely,

John Doe

That is, the indentation, words, and spacing never change. If done on a typewriter, it all has to be typed every time, but with a Word Processor, the signature area can be stored on disk under an easy-to-remember name, such as "SIGN", and then imbedded at the end of every letter. In ALLWRITE, this can be done as follows:

--- the 'SIGN' file ---

;sk3  
;in40

Sincerely,  
;sk 2  
John Doe

--- any letter ---

... to hear from you soon.  
;imbed sign

Obviously, it takes much less typing to say ";im sign" than to space down and to the right, and then type that all in every time. If you're really lazy, you could even define a Soft Key to stand for ";im sign", which would reduce 39 keystrokes to two.

The concept can be extended to what is often called "boilerplate". This term refers to stock sentences, paragraphs, or whole sections of standard material that never change, or need only minor modification when used. If a proposal is typed, the boiler plate (contract terms, product description, etc.) would have to be typed in every time. Even if done by a secretary, it would take time and likely include 'typos' that would have to be proof-read and corrected later. However, if boilerplate is entered into a Word Processor and then proof-read, it never has to be typed or proofed again. All that's needed to use it is to 'imbed' it, just as we did for the signature. If it has to be modified, a copy of the stock material can be modified quickly for the one-time situation.

### Using a Spelling Checker

I know two people who can glance at a sheet of paper, and without reading it, instantly point to spelling errors. I don't know how they do it, and don't have their gift. Fortunately, I have access to a computer program that can do the same thing for me. When I finish writing this chapter, I'll just press the Spelling checker key and Electric Webster (EW) will show me every word it doesn't recognize. This will take about a minute, and then EW will let me correct the words that are wrong, "teach" the words that it doesn't know, and even look words up in its dictionary to make sure I've typed the corrections correctly! The time the correction step takes is limited by how fast I can type, plus about 15 seconds to re-write the corrected document to disk.

Oh, yes: Electric Webster can also insert conditional hyphens for me, and then ALLWRITE will use or discard those conditional hyphens when it formats and prints the article. What do I have to do to make all this happen? Just press the letter "Y" when asked if hyphenation is desired. To carry this kind of convenience even further, Electric Webster has another option that checks grammar, usage, and punctuation. I don't think I want my kids to use these things until they've learned to spell and write correctly on their own, but tools like this are such time-savers that it's hard to ignore them: they are to language what calculators are to math.

Will all spelling errors be corrected? Probably not, because I may tell the checker to accept a misspelled word. Even though the checker called it to my attention, I may be mistaken [sic] in how I believe something is spelled. But, that error will be my fault, not the computer's.

**Summary**

The power of a computer lies not just in its ability to store information, but in its flexibility to retrieve and restructure that information afterwards. Properly used, a computer becomes an extension of the human mind, providing the mass storage of a dull facts, the rapid retrieval of them, the lightning fast and accurate calculations, and the presentation of the results in visual form. This gives us time to think and be creative, without becoming bogged down in the mechanics of storing and searching, counting and writing, correcting and revising. People are very good at being creative, and computers are very good at rot task.

Its seems like a fair way to divide the work.

\* \* \* \* \*

This ends our presentation of "ALLWRITE". We hope you will enjoy it as much as we do, and that it will meet your word processing requirements for many years to come.